De l'usage des méthodes bas niveau pour la recherche d'image par le contenu

Jérôme da Rugna

THÈSE

présentée en vue de l'obtention du titre de

Docteur de l'Université Jean Monnet Saint-Étienne

spécialité

Informatique : Image

par

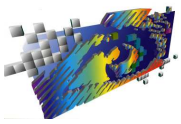
Jérôme Da Rugna

De l'usage des méthodes bas niveau pour la recherche d'images par le contenu

---

Composition du Jury

Christine Fernandez-Maloigne	Rapporteur
Jean-Michel Jolion	Rapporteur
Patrick Lambert	Examinateur
Simone Santini	Examinateur
Hubert Konik	Examinateur
Bernard Laget	Directeur de thèse
Alain Trémeau	Invité



Soutenue publiquement le
14 Décembre 2004

Laboratoire
LIGIV

REMERCIEMENTS

Je tiens à remercier avant tout Hubert Konik qui a dirigé cette thèse tout au long de ces années. Il a été l'encadrant que je recherchais et il a su me motiver, orienter mes recherches et permettre l'aboutissement de ce travail. Pour tout cela et pour tout le reste, je le remercie.

Je remercie les rapporteurs de cette thèse, Christine Fernandez-Maloigne et Jean-Michel Jolion, pour la rapidité avec laquelle ils ont lu mon manuscrit et l'intérêt critique qu'ils ont porté à mon travail. Leurs commentaires et leurs remarques ont permis non seulement d'améliorer le document initial mais aussi de me donner de l'inspiration et de la motivation pour ma recherche future.

Je me réjouis de la présence de Mr Patrick Lambert à mon jury . Je l'en remercie vivement.

Je remercie Simone Santini pour tous les kilomètres parcourus et les aéroports visités. Il m'a accueilli, avec Amarnath Gupta, durant six mois fructueux à l'université de San Diego. Qu'ils en soient remerciés ici.

Aux membres du laboraroire LIGIV, passés ou présents, je ne peux que leur être généreusement reconnaissant de leur efforts, petits ou grands, pour m'accompagner tout au long de ce travail de thèse. Que chacun se reconnaisse dans ses remerciements qu'il ne m'est pas possible d'exprimer entièrement ici.

Je ne peux oublier de remercier Mr Bernard Laget, mon directeur de thèse, et Mr Alain Trémeau, le directeur du laboratoire LIGIV, sans qui cette thèse n'aurait pu prendre forme.

Je remercie en substance Anne Catherine pour le soutien immodéré qu'elle m'a apportée ces derniers mois.

À vous tous et à ceux que j'ai oublié, un véritable remerciement se donne de vive voix, sachez que ce sera fait!

INTRODUCTION

L'origine de ces travaux de thèse est intervenue dans une période de plein essor pour les systèmes de recherche d'images par le contenu. La multitude des activités, traduite soit par des publications soit par des démonstrations en ligne, avaient fait naître de nombreux espoirs. Véritablement, quelle était la réelle ambition de ces systèmes ? D'appréhender la sémantique même d'une image uniquement à partir de son contenu. Habituellement, ces systèmes proposaient à l'utilisateur de rechercher une image parmi d'autres selon certains critères préétablis. Pour cela, trois étapes permettaient généralement une stratégie de recherche:

- L'interface proposée à l'utilisateur. Celle-ci doit lui permettre de signifier au système ce qu'il attend et lui proposer en retour les résultats de sa recherche.
- L'extraction des données. Les paramètres numériques permettant de décrire les images sont extraits de celles-ci et archivés. Ils renseignent sur le contenu exclusivement bas-niveau et reposent sur des méthodes issues de l'analyse et du traitement d'images.
- La gestion des connaissances. Le système doit traduire la requête sous-jacente en exploitant au mieux les informations relatives à chaque image.

Néanmoins, face à l'engouement de la communauté pour ces systèmes, les utilisateurs finaux ont opposé leurs complexités et leur philosophie de la recherche d'images. Le fossé sémantique entre leurs attentes et les capacités réelles de ces sytèmes n'a pas permis à ces derniers de se positionner en acteur incontournable de la recherche multimédia. Devant la nécessité d'une remise en question, il est donc devenu inéluctable de converger vers une approche plus réaliste, où la sémantique parait difficilement accessible uniquement à partir du seul contenu des images. Non seulement par manque de méthodes universelles mais aussi par essence même. En effet, la sémantique d'une image n'est pas seulement descriptible à partir des objets qui la composent mais aussi naturellement relative à l'utilisateur, depuis ses états d'âme jusqu'à sa mémoire individuelle et contextuelle.

Devant ces lacunes, quelle pierre peut-on toutefois apporter à l'édifice ? Sans aucun doute et en premier lieu, tester la validité et approfondir la réelle capacité de tous les outils mis en

jeu, même les plus basiques. En effet, toutes les phases aboutissant à une similarité se doivent d'être analysées et évaluées de manière objective, afin de s'assurer que les axiomes que l'on suppose sur eux soient justifiés. Par exemple, il est classique de filtrer l'information de l'image par une étape de segmentation supposée partitionner l'image en entités conjecturables. Cette étape devient ainsi le coeur du mécanisme d'extraction de l'information visuelle d'une image. Avant même de critiquer les descripteurs fondés sur elle, il est alors nécessaire de savoir si la segmentation elle-même répond déjà à l'espoir placé en elle. De ce constat, nous avons axé une importante partie de nos travaux sur l'évaluation objective des méthodes de segmentation. En particulier, nous présenterons un protocole d'évaluation de ces méthodes dans un contexte de recherche d'images par le contenu.

Pourtant, se contenter de remettre en question les différentes étapes de la recherche de similarité ne peut être une finalité en soi. Il faut aussi, à partir de ces conclusions, faire émerger de nouvelles pistes ou construire de nouveaux outils sous le couvert des réelles capacités de chacune des phases. La portée de ces outils se doit alors d'être réaliste : il est essentiel que l'expert puisse parfaitement maîtriser les différentes informations qui en découlent. La recherche par similarité doit pouvoir s'appuyer sur des descripteurs intelligibles et robustes. L'apport d'"outils ciblés" face à des "indexeurs généralistes" trouve alors pleinement sa justification. On entend par "outil ciblé", une analyse simple de l'image qui permet d'obtenir une information bas niveau. En parallèle du peu d'informations sémantiques contenues dans la description, une grande stabilité et une complète maîtrise peuvent être envisagées. Dans ce contexte, nous présenterons une extraction bas niveau de régions couleurs émergentes d'une image.

Dans ce même contexte, est-il possible de capitaliser une somme de connaissances basiques autour de chaque image, avant même de réaliser les recherches de similarité ? Et ainsi utiliser ce que l'on appelle communément les méta-données. Avec certes une confiance relative, il est possible par exemple de détecter si une image est d'intérieure ou d'extérieure, ou si elle correspond à un gros plan ou un plan éloigné. Cette information doit permettre de guider l'extraction des connaissances et leur gestion. Nous proposerons alors d'extraire une méta-donnée qui influence fortement notre perception de l'image : le flou. Cet indice visuel exprime entre autre une notion de profondeur qui permet de délimiter rapidement les différentes sensations de l'image. Cette fragmentation des éléments participe positivement à la reconnaissance de l'image et oriente la focalisation vers certaines zones particulièrement riches en information.

Au final, quand bien même il est possible de se reposer sur des descripteurs de bas niveau maîtrisés, sommes-nous à ce moment-là capables de les exploiter au mieux ? Les systèmes classiques ne brident-ils pas l'utilisateur face à toute l'information accessible ? Par exemple, et ceci constituera un chapitre de cette thèse, comment tirer profit de toute l'information intrinsèque des histogrammes couleur sans se limiter à un simple calcul de distance entre deux histogrammes.

Ainsi nous introduirons un modèle algébrique de recherche d'images avec une application sur les histogrammes. Ces prémices constitueront une première réponse au besoin naissant de modèle spécifique pour la gestion de bases d'images.

Les contributions de cette thèse, intitulée *"De l'usage des méthodes bas niveau pour la recherche d'images par le contenu"*, vont être scindées en trois parties, articulées autour de la figure 1.

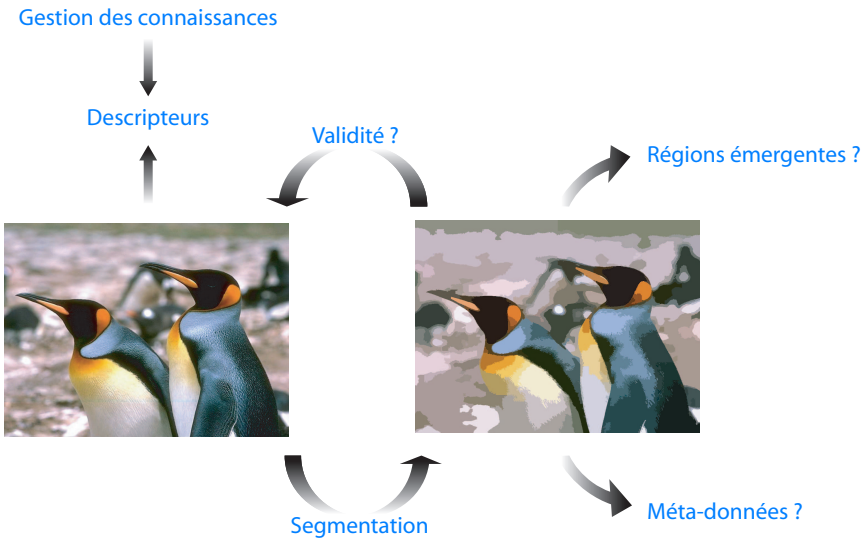


Fig. 1 – *Contributions de la thèse*

- La première partie dresse un état de l'art succinct du domaine de la recherche d'images par le contenu de ces dernières années. Il met en exergue les difficultés rencontrées face aux problèmes réels et le fossé sémantique, qui semble infranchissable, entre la demande des utilisateurs et les réelles capacités des systèmes. Il permet de recentrer notre travail de thèse et présente les motivations des contributions apportées.
- La seconde partie débute par une présentation des différentes méthodes de segmentation usuellement utilisées dans les systèmes de recherche par le contenu. Devant la difficulté à les juger de façon objective, elle se poursuit par la mise en place d'un protocole d'évaluation pour mesurer leur stabilité et la confiance que l'on peut leur donner.
- La troisième partie présente les développements que nous avons introduits pour la mise en place de quelques outils ciblés pouvant améliorer la recherche par le contenu dans des objectifs bien spécifiques. Le premier chapitre expose ainsi un détecteur de flou qui isole dans une image les différents plans de focalisation. Dans une même démarche de filtrage intelligible de l'information de l'image, le deuxième chapitre est consacré à une méthode

d'extraction automatique de régions couleurs émergentes ainsi qu'à son rôle possible dans une recherche de similarité. Finalement, un dernier chapitre introduit un langage permettant d'interroger une base d'histogrammes de manière simple tout en conservant toute l'information intrinsèque contenu dans ce descripteur classique, jetant les bases d'un modèle opérationnel de gestion de connaissances.

❖ En définitive, la conclusion présente les principaux apports de ce travail et introduit certaines perspectives.

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

Première partie

Recherche d’images par le contenu

Chapitre 1

THE HOLY GRAIL

Sommaire

Rechercher une image parmi d'autres, tel est l'enjeu de la recherche d'images par le contenu. Ce chapitre se propose d'établir un état de l'art concis de toutes les techniques mises en œuvre ces dernières années tant sur la description basique d'une image par sa couleur, sa forme ou sa texture que sur l'intégration de ces données dans un véritable système de recherche d'images par le contenu.

1.1 Qu'est-ce qu'une photographie ?

Une large gamme d'outils et de procédés permet de créer une image. Tout d'abord, une image peut être la résultante de la capture instantanée d'une scène donnée, communément appelé photographie. Elle peut peut aussi être produite par d'autres moyens : synthèse d'images, assemblage de diverses images (mosaïque, cartes postales, ...), par procédé technologique comme les scanners médicaux. Une image peut aussi résulter de toute altération d'une précédente image, par de simple outils de retouches d'images par exemple.

Sans vouloir spécifier explicitement le sens réel donné au mot photographie, nous allons nous intéresser plus précisément à la catégorie que constitue l'ensemble des images dites "photographies".

Un jour, à une heure bien précise de la journée, une personne, photographe amateur ou bien professionnel, a pris une photographie de la scène qu'elle regardait, et il en est résulté l'image numérique que nous désirons étudier. Malheureusement, ces seules informations sont nettement insuffisantes pour le faire. Si l'art de la photographie consiste, en un sens, à donner toute l'expression possible à une image, celle-ci doit exprimer le sentiment que l'auteur ressentait à ce moment là. Il en est tout autrement pour toutes les autres images, la plus grande majorité d'ailleurs. De ces classiques photographies de vacances, excepté l'auteur et parfois son entourage, qui peut bien se targuer de pouvoir en saisir le sens premier ? Hors contexte, une image n'est le plus souvent qu'un très mauvais reflet de la scène qu'elle représente.

Arrêtons nous sur la notion de contexte, qu'il est possible de séparer en deux phases:

- Le moment de la création, l'instant où la photographie est prise.
- Le moment de son analyse, l'instant où la photographie est regardée.

Si le contexte est statique au moment de la création d'une image, il n'en est pas du tout de même quant au visionnage d'une image. L'appréciation d'une image est très subjective et est intimement liée à des facteurs émotionnels propres à chacun.

1.1.1 La création

Différents points peuvent être distingués:

- L'auteur
  La personne qui se tient derrière la caméra est forcément l'acteur le plus important dans le processus de fabrication d'une image. C'est celui qui déclenche la prise d'image au moment où il le juge opportun. C'est celui qui cadre, qui choisit la scène, son orientation, sa distance aux objets...
- La technique
  Le type de caméra, ou d'appareil photographique, le type de pellicule, la résolution et

l’illuminant, dans le cas des appareils numériques, l’utilisation ou non d’un flash, sont autant de facteurs parmis d’autres, purement techniques, qui influent sur l’image elle-même, et notamment sur sa représentation numérique. En cela, la technique fait partie à part entière du contexte. Il s’agit a priori du contexte le plus quantifiable, même s’il est difficile de simplement calculer ou d’approximer ne serait-ce que les caractéristiques de l’illuminant [Tominaga et Wandell, 2002].

- La scène
  Sans doute aussi importante que l’auteur, la scène (dans la globalité) y est intimement liée. En effet, celle-ci se déroule sous les yeux de l’auteur et, malheureusement pour celui qui désire comprendre l’image issue de la scène, le facteur temporel influe considérablement. De même qu’il est parfois tendancieux d’extraire certaines phrases de leur contexte. Certaines informations relatives au contexte non présentes dans la photographie sont primordiales à la compréhension de l’image et de ce qu’elle représente.

1.1.2 La perception

Si l’on admet que la mémoire de l’auteur lui permet de se souvenir de chacun de ses clichés (ce qui est loin d’être toujours le cas d’ailleurs), le sens donné à une image est fortement dépendant de la personne qui la visionne. Les éléments qui influent sur cette dernière peuvent être par exemple :

- Ce qu’attend le spectateur.
- Ce qu’il a déjà vu.
- Ce qu’il connaît du contexte de la création.
- L’attention qu’il porte à l’image.
- …

Fig. 1.1 – *Où la connaissance de la page suivante importe. . .*

Une image, si elle a au départ un seul et unique seul sens, peut représenter une multitude de perceptions différentes suivant la personne qui la regarde et les connaissances qu’il possède. Par

Fig. 1.2 – *L'image complète. . .*

exemple, la perception de l'image 1.1 va être fortement influencée par la perception de l'image 1.2.

De tout cela, rechercher une image consiste, suite à une requête donnée par un utilisateur lambda, à fournir à celui-ci toutes les images que cet utilisateur lambda percevra justement comme répondant à sa requête. La communauté scientifique, tant celle des systèmes d'informations que celle du traitement d'images, a tenté d'apporter un certains nombre de réponses à cette problématique [Smeulders *et al.*, 2001].

1.2 La recherche d'images par le contenu : genèse

Ces dernières années, l'ère du numérique aidant, toutes ces photographies arrivent massivement sur nos ordinateurs. Pour autant, le temps où ceux-ci n'étaient pas capable d'afficher une image n'est pas si lointain. Avant que des millions d'images circulent sur internet et accentuent le phénomène, les problèmes, tout d'abord d'archivage, puis de consultation et finalement de recherche d'images se sont posés. Le mot indexation est arrivé via les spécialistes de bases de données, qui géraient justement les bases d'images. En effet, très rapidement, ces derniers ont été demandeurs d'index pour classifier les images et pour accélérer les recherches dans ces bases. L'indexation d'images prit donc essor dans les années 80 via les mots clés. C'est ce que nous verrons après avoir brièvement survolé quelques applications qui ont font naître cet engouement.

1.2.1 Les applications et le domaine des images

La liste d'applications possibles de la recherche d'images par le contenu est immense. Certaines paraissent déjà indispensables, d'autres le deviendront quand les systèmes seront performants. Citons les plus importantes, accompagnées d'une petite illustration.

- Architecture - Retrouver des bâtiments ou des aménagements intérieurs,...
- Musée, galerie - Explorer et rechercher des peintures similaires,...
- Vidéo, films - Retrouver des scènes, des parties d'un film,...
- Militaire - Repérer toutes les images avec un char ennemi, ...
- Agences photographiques - Rechercher des photos de telle ou telle célébrité, ...
- Surveillance - La scène est-elle la scène d'un vol de voiture ?, ...
- Photographe amateur - Où est la photo de mes enfants des vacances 1992 ? , ...
- Moteur de recherche - Chercher sur internet une photo libre de droit pour illustrer un document, ...

Ces dernières années, nous pouvons souligner que l'arrivée des appareils photographiques numériques et des grandes capacités de stockage, le tout à faible coût, a permis le développement grandissant de demande de gestion de bases de données images. En premier lieu, ce fût le cas pour les agences photographiques où la grande majorité de leur images sont maintenant directement issues d'appareils numériques, le reste étant aussi numérisé. Remplacer le rôle de l'archiviste n'est pas encore d'actualité mais les demandes sont fortes pour supplanter l'homme pour la recherche d'images. Le grand public aussi se voit proposer des caméras numériques. Nombreux sont les amateurs possédant des bases d'images de bonne taille, qu'ils partagent sur internet ou non. Ce sont sans doute les clients de demain d'un système performant de gestion et de recherche dans les bases d'images.

Brièvement, introduisons la notion de domaine des images. Nous entendons par domaine des images, l'ensemble qui englobe toutes les images d'une application donnée. Par exemple, le moteur de recherche sur internet se place dans un modèle généraliste, le domaine est donc l'ensemble des images possibles. Par contre, l'architecte recherchant une construction bien précise se place dans un domaine beaucoup plus restrictif : celui de sa base de donnée architecture. C'est une notion virtuelle mais qui est très importante. Le domaine dicte en partie la stratégie de recherche. Suivant le domaine des images, la mesure de similarité, par exemple, n'est pas la même.

1.2.2 Les mots clés : un descripteur historique

Dès les prémices des bases de données images, les spécialistes ont décrit celles-ci via des attributs textuels. Chaque image est ainsi annotée par une série de mots, chacun de ces mots

décrivant un objet ou un sens de la scène. De nombreux travaux ont été faits sur la gestion des images grâces aux mots clés [Cardenas *et al.*, 1993, Srihari, 1995], pour deux principales raisons :

- Historique : bien avant de numériser les photographies, les mots clés servaient à indexer les images, repérées par exemple par un numéro dans la base. Une fois la recherche terminée, l'archiviste allait récupérer physiquement les images.
- Technique : la capacité informatique permettant d'extraire l'information d'une image est très récente. Tous les systèmes mis en place avant ces dernières années sont donc tous uniquement basés sur les mots clés.

La figure 1.3 illustre un exemple de classification d'images par mots clés.

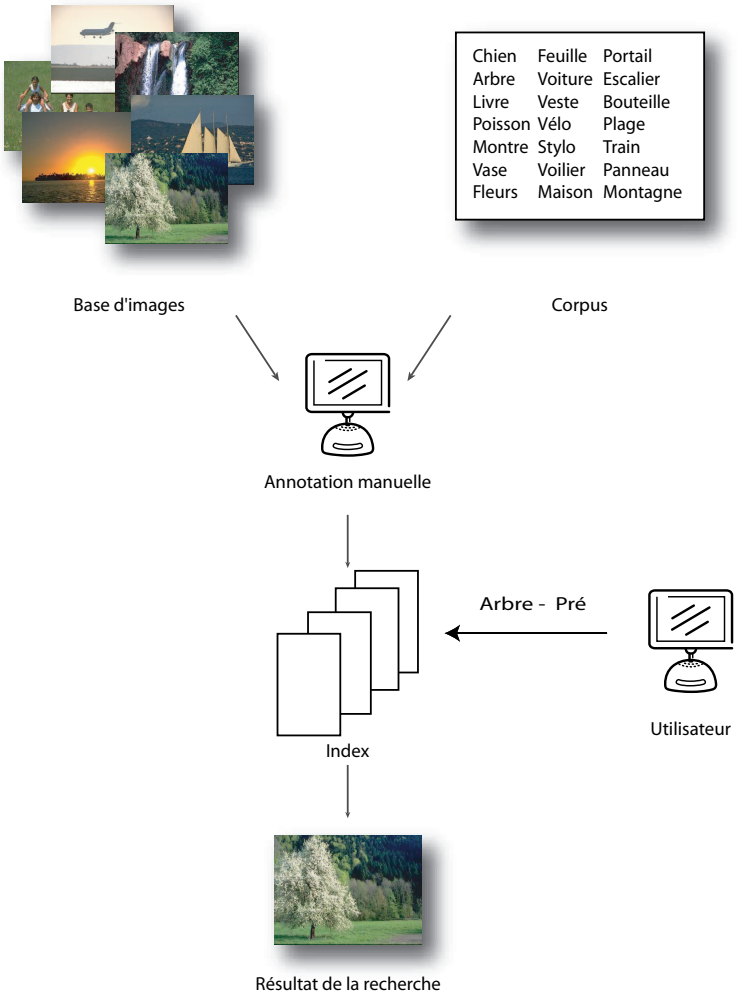


Fig. 1.3 – *Indexation par mots clés*

1.2.3 Où le traitement d'images apparaît

Afin de pallier l'évident problème d'un corpus, le piège de l'exhaustivité, utiliser le contenu pixellique d'une image pour en extraire une information sémantique semble justifié. La figure 1.4, sur laquelle nous reviendrons plus en aval dans ce manuscrit, montre le schéma général d'une recherche par le contenu. L'extraction de paramètres numériques à partir d'une image a vu éclore une imposante diversité d'approches. Sans citer l'intégralité des nombreuses méthodes, nous allons tout d'abord relever un certains nombres de paramètres classiques et de mesures de similarités correspondantes, tant pour l'aspect couleur, que forme ou texture. Ensuite, nous nous attacherons à décrire les différents modèles qui permettent d'aboutir à un véritable système de recherche par le contenu. Finalement, nous listerons différents systèmes de recherche qui ont marqué la communauté.

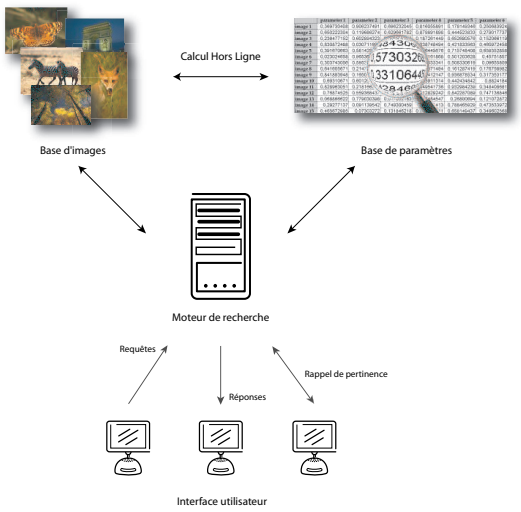


Fig. 1.4 – *Systèmes de recherche d'images par le contenu*

1.3 Les signatures couleurs

La recherche d'images via la couleur est sans aucun doute le sujet qui a pris le plus grand essor ces dernières années. De nombreux travaux ont vu en effet le jour quant à l'utilisation de l'information couleur et sur l'analyse des différents modèles. Après avoir succinctement introduit diverses représentations de la couleur et leurs distances associées, nous décrirons les principales approches couleur concernant la recherche de similarité.

1.3.1 Représentation de la couleur

La couleur peut être vue comme une information trichromatique des stimuli du spectre visuel. Sans entrer dans une analyse physiologique de la perception humaine, décrivons brièvement les principaux espaces couleurs que nous avons étudiés et utilisés durant ce travail.

Tous les espaces introduits [Colantoni, 2003] sont inclus dans la librairie IVLIB développée au sein du laboratoire LIGIV par P. Colantoni principalement et moi-même [1] . Comme l'expliquent les auteurs dans [Colantoni et Trémeau, 2003], la visualisation des images à traiter dans différents espaces couleur peut fournir des éléments fondamentaux quant à l'attente que nous pouvons avoir de certains algorithmes de traitement d'images. De plus, cette visualisation peut permettre de comprendre les raisons de l'échec d'une méthode dans l'espace considéré.

1.3.1.1 L'espace initial : RGB

Dans la quasi totalité des applications actuelles, l'espace d'acquisition et de sauvegarde des images est basé sur l'espace couleur RGB. L'acquisition utilise cet espace pour des raisons techniques évidentes dues aux capteurs de type RGB. Cela influt évidemment sur l'espace couleur utilisé pour sauvegarder l'image, car nous ne connaissons pas, dans de nombreux cas, les caractéristiques permettant une autre modélisation exacte. Nous reviendrons sur cet aspect plus en aval dans ce manuscrit. Il existe différents types d'espaces RGB, qui dépendent du matériel employé : choix des longueurs d'ondes et des primaires par exemple. En fait, dans très peu de cas, nous pouvons considérer que l'espace RGB correspond à l'espace additif défini dans CIE [2] 1931 [CIE, 1971], ie sur les primaires rouge (*700 nm*), vert (*546.1 nm*) et bleu (*435.8 nm*).

Un autre espace couleur dépendant du matériel, principalement pour l'impression des couleurs, est l'espace Cyan, Magenta, Yellow. Cet espace soustractif est défini comme le triplet $(C,M,Y) = (1-R,1-G,1-B)$.

La distance couleur associée à ces espaces est la distance euclidienne.

Le triangle de Maxwell représenté dans l'espace RGB(figure 1.5) est le triangle reliant les trois couleurs pures Rouge, Vert et Bleu. Il nous permettra ainsi de suivre les différentes distorsions dues aux changements d'espaces couleur.

1.3.1.2 L'espace charnière : XYZ

Le système de primaires XYZ est l'espace de base dans le monde de la couleur. Tous les espaces couleurs ont un lien avec l'espace XYZ défini en 1931 et affiné en 1964 par la CIE

1. La librairie supporte différentes possibilités de calibrage (par approximation ou non) et inclut un important jeu de primaires et de blancs de référence. L'affi chage en 2 ou 3 dimensions des différents espaces est accompli avec le logiciel ColorSpace développé au LIGIV par P. Colantoni

2. Commision internationale de l'éclairage. http://www.cie.co.at/cie/

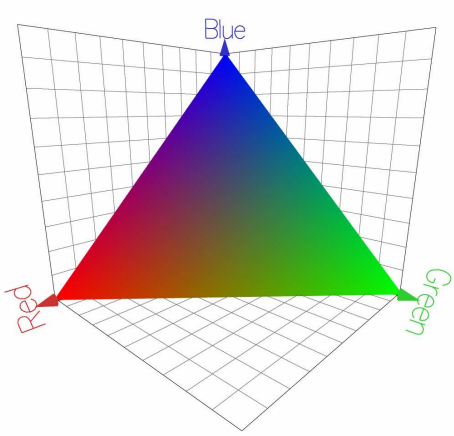


Fig. 1.5 – *Triangle de Maxwell dans l'espace* RGB

[CIE, 1971]. Le passage de l'espace RGB CIE 1931 à l'espace XYZ CIE 1931 se fait par la formule :

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 2.7690 & 1.7518 & 1.1300 \\ 1.0000 & 4.5907 & 0.0601 \\ 0.0000 & 0.0565 & 5.5943 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1.1}$$

Les vecteurs de la matrice correspondent aux coordonnées des primaires rouge, vert et bleu. Bien sûr, si le système de primaires n'est pas le CIE RGB, comme dans le cas du RGB des normes NTSC ou PAL [3] par exemple, la matrice de passage est à recalculer.

La distance couleur associée à l'espace XYZ est la distance euclidienne.

L'espace xyY, appelé diagramme de chromaticité, est communément utilisé pour représenter les coordonnées trichromatiques. Cet espace présente l'avantage de séparer l'information chromatique xy de l'information achromatique Y .

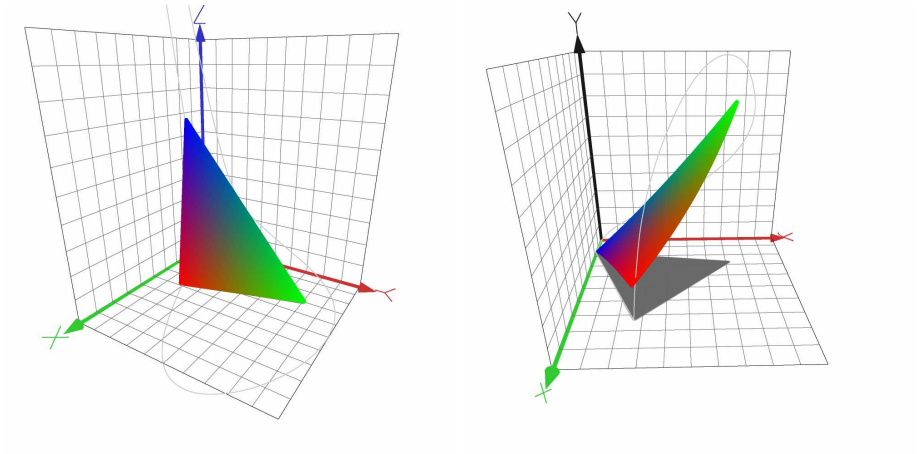
xyY est obtenu via la formule :

$$x = \frac{X}{X+Y+Z} \qquad y = \frac{Y}{X+Y+Z} \qquad Y = Y \tag{1.2}$$

1.3.1.3 Les espaces chrominance - luminance

Une autre possibilité intéressante, dans l'optique d'un traitement de la couleur, est de décorréler la chrominance de la luminance. Ainsi l'information chrominance est portée sur deux axes, et l'information luminance sur le troisième. Nous pouvons ici citer différents espaces, comme $YCbCr$, YIQ, YUV ou encore $AC1C2$.

3. Standards de codage pour la retransmission télévisuelle



(a) Espace XYZ (b) Espace xyY

Fig. 1.6 – *Triangle de Maxwell dans les espaces XYZ et xyY*

$YCbCr$ est obtenu via la formule :

$$\begin{cases} Y &= 0.299 \times R + 0.587 \times G + 0.114 \times B \\ Cb &= -0.169 \times R - 0.331 \times G + 0.500 \times B \\ Cr &= 0.500 \times R - 0.418 \times G - 0.082 \times B \end{cases} \tag{1.3}$$

YIQ la version de $YCbCr$ correspondant au standard NTSC, est obtenu via la formule :

$$\begin{cases} Y &= 0.299 \times R + 0.587 \times G + 0.114 \times B \\ I &= 0.596 \times R - 0.274 \times G - 0.322 \times B \\ Q &= 0.212 \times R - 0.523 \times G + 0.311 \times B \end{cases} \tag{1.4}$$

On peut noter que le calcul pour ces deux espaces de la composante Y est similaire, seules les composantes chromatiques diffèrent.

1.3.1.4 Les espaces par partitions

De nombreux travaux, concernant les modèles d'espaces couleur, se sont orientés vers un partitionnement géométrique (ou pseudo-géométrique) de l'espace couleur. Munsell [Munsell, 1946] introduit en 1946 un système basé sur une décomposition Teinte/Saturation/Luminance. Une discrétisation avec une dynamique propre sur chacun des trois axes permet de classer chaque couleur, comme l'illustre la figure1.8. Ce sytème est régi par deux lois fort intéressantes :

- Si deux couleurs sont non différentiables, alors elles ont les mêmes coordonnées.
- Si deux couleurs sont identiquement différentiables d'une troisième, alors elles sont à la même distance de celle-ci.

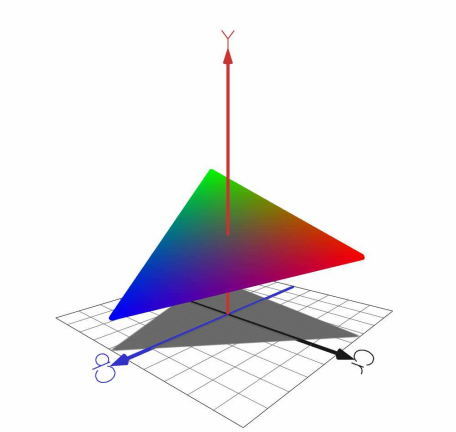


Fig. 1.7 – *Triangle de Maxwell dans l'espace* $YCbCr$

Cet espace est encore fortement utilisé dans de nombreux domaines, pour la classification visuelle des couleurs: il est la source notamment de la majorité des Color Chart [4].

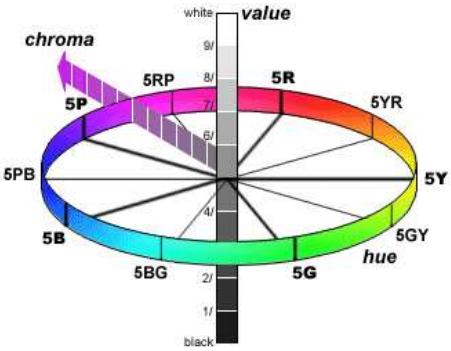


Fig. 1.8 – *Système de Munsell*

Une autre modélisation géométrique communément utilisée est basée sur les trois composantes Teinte, Saturation et Luminance. La différence entre ces espaces, comme HSV (Hue Saturation Value) ou HSI (Hue Saturation Intensity) provient des équations utilisées pour calculer littéralement les trois composantes.

Détaillons l'espace HSV qui, dans la suite de ce document, sera utilisé plusieurs fois. La méthode employée pour calculer l'espace HSV est illustrée par l'algorithme 1. Ainsi calculée,

4. Panel de couleurs distinctes servant souvent de référence à de nombreux travaux sur la couleur (calibrage d'impression ou d'acquisition...). Les plus connus sont les color chart de Munsell et Mac Beth

la dynamique de l'espace est :

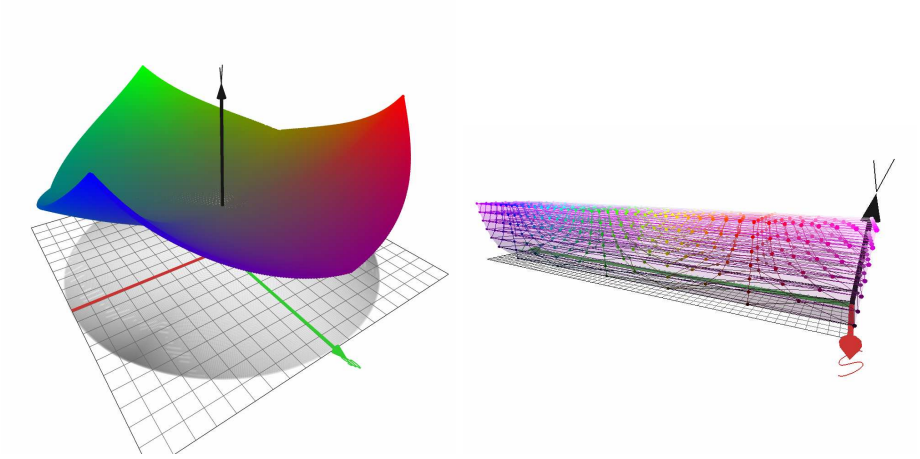
$$\begin{cases} H & \in & [0..6[\\ S & \in & [0..1] \\ V & \in & [0..1] \end{cases}$$

Il faut bien remarquer que la composante H est une composante cyclique, ie $H = 6$ n'est pas réalisable du fait de la circularité, ce cas là correspondant précisément à $H = 0$. Ainsi la distance de teinte entre deux valeurs H_0 et H_1 peut s'écrire dans cet espace :

$$\Delta H = \min (\quad \|H_0 - H_1\| \quad , \quad 6 - \|H_0 - H_1\| \quad) \tag{1.5}$$

Dans le cadre d'indexation d'objets par exemple, la séparation de la composante teinte du reste de l'information peut être très avantageuse. En effet, nous désirons souvent comparer la couleur de l'objet avec d'autres références. En cela, comparer uniquement la teinte correspond à une certaine attente visuelle d'une similarité couleur.

De plus, la teinte est un bon invariant à de nombreux effets colorimétriques, comme les changements de luminance ou de saturation. Néanmoins, du fait de la transformation non linéaire depuis RGB il peut résulter des imprécisions numériques, notamment dans les cas de faible saturation.



(a) Triangle de Maxwell (b) Cube RGB

Fig. 1.9 – *Espace couleur* HSV

1.3.1.5 Les espaces décorrélés

L'espace couleur $X_1X_2X_3$ est le résultat d'une décomposition en 3 axes décorrélés, grâce à la méthode de Karhunen - Love, dans laquelle la corrélation entre les différents axes est inexistante. Nous pouvons ainsi traiter indifféremment chaque axe sans risque de perte d'information.

Algorithme 1: Transformation RGB en HSV

Données : Triplet RGB

si $R > G$ **alors**
  $Max = R$; $Min = G$; $position = 0$
sinon
  $Max = G$; $Min = R$; $position = 1$

si $Max < B$ **alors** $Max = B$; $position = 2$
si $Min > B$ **alors** $Min = B$
$V = Max$
si $Max \neq 0$ **alors**
  $S = \frac{Max - Min}{Max}$
sinon
  $S = 0$

si $S \neq 0$ **alors**
  si $position = 0$ **alors**
    $H = \frac{1 + G - B}{Max - Min}$
  sinon
    si $position = 1$ **alors**
      $H = \frac{3 + B - R}{Max - Min}$
    sinon
      $H = \frac{5 + R - G}{Max - Min}$

Résultat : Triplet HSV

Néanmoins, l'inconvénient majeur est que la décomposition est unique à toute image: l'espace couleur est lié à l'image elle-même. Le second inconvénient est le coût algorithmique important de la décomposition. Pour pallier à ce problème, Ohta [Ohta *et al.*, 1980] propose l'espace $I_1I_2I_3$ dont les coefficients de la matrice de passage entre RGB et $I_1I_2I_3$ sont prédéfinis. En effet, dans le cas des images dites "naturelles", l'axe principal se confond avec l'axe de luminance et le système d'axe $X_1X_2X_3$ reste relativement stable quelles que soient les images. Ainsi $I_1I_2I_3$ est une approximation de la décomposition de Karhunen - Love dans le cas d'images naturelles.

La distance couleur associée à ces espaces est principalement la distance euclidienne.

1.3.1.6 Les espaces perceptuels

Le principal défaut des précédents espaces est la non linéarité par rapport à la vision humaine. En effet, la même différence colorimétrique entre deux couples n'a pas le même sens suivant la position dans le cube RGB. C'est de ce constat qu'est apparue l'étude des espaces plus proches de la perception humaine dits uniformes, tels que $L^*a^*b^*$ ou $L^*u^*v^*$. Notons que l'aspect "perception humaine" n'est que local, c'est-à-dire n'est approché que dans le cas de couleurs proches. Dans le cas où deux couleurs seraient fortement éloignées, la distance dans l'espace $L^*a^*b^*$ n'a pas plus de "sens" que dans d'autres espaces. Il faut quand même noter que la volonté d'uniformité (par rapport aux ellipses de MacAdam [Brown et MacAdam, 1949] notamment) n'est pas vraiment atteinte mais est beaucoup plus satisfaisante que pour les autres espaces couleur.

X_0, Y_0, Z_0 représente l'illuminant de référence. La transformation non linéaire, qui permet de passer de XYZ à $L^*a^*b^*$ est alors la suivante :

$$\begin{cases} L^* &= \begin{cases} 116\left(\frac{Y}{Y_0}\right)^{\frac{1}{3}} - 16 & pour \quad \frac{Y}{Y_0} > 0.008856 \\ 903.3\left(\frac{Y}{Y_0}\right) & pour \quad \frac{Y}{Y_0} \leqslant 0.008856 \end{cases} \\ a^* &= 500\left[f\left(\frac{X}{X_0}\right) - f\left(\frac{Y}{Y_0}\right)\right] \\ b^* &= 200\left[f\left(\frac{Y}{Y_0}\right) - f\left(\frac{Z}{Z_0}\right)\right] \end{cases} \tag{1.6}$$

où :

$$f(x) = \begin{cases} x^{\frac{1}{3}} & pour \quad x > 0.008856 \\ 7.787x + \frac{16}{116} & pour \quad x \leqslant 0.008856 \end{cases} \tag{1.7}$$

Ainsi :

- ΔL^* représente la différence de luminance
- Δa^* représente la différence de chromatique rouge-vert
- Δb^* représente la différence de chromatique jaune-bleu

Dans cet espace, la distance couleur est finalement donnée par :

$$\Delta E^*_{ab} = \sqrt{\Delta L^{*2} + \Delta a^{*2} + \Delta b^{*2}} \tag{1.8}$$

On considère alors qu'une distance inférieure à 1 indique deux couleurs visuellement non différentiables. Dans de nombreux cas, on augmente le seuil de non différentiation à 2 voire 3 [Trémeau *et al.*, 2004] .

Notons aussi une autre distance possible dans les espaces $L^*a^*b^*$ ou $L^*u^*v^*$. Il faut d'abord définir la chrominance et l'angle de teinte, dont la formule pour le cadran positif est :

$$C^*_{ab} = \sqrt{a^{*2} + b^{*2}} \quad \text{et} \quad h_{ab} = \arctan\left(\frac{b^*}{a^*}\right) \tag{1.9}$$

Cela définit en fait l'espace connu sous le nom LHC. Notons : $\Delta C^*_{ab} = C^*_2 - C^*_1$ et $\Delta h_{ab} = h_2 - h_1$

L'écart angle de teinte est défini par:

$$\Delta H^*_{ab} = 2\sqrt{C^*_1.C^*_2}.\sin\left(\frac{\Delta h_{ab}}{2}\right) \tag{1.10}$$

En définitive, le formule de l'écart couleur peut s'ecrire :

$$\Delta E^*_{ab} = \sqrt{(\Delta L^*_{ab})^2 + (\Delta C^*_{ab})^2 + (\Delta H^*_{ab})^2} \tag{1.11}$$

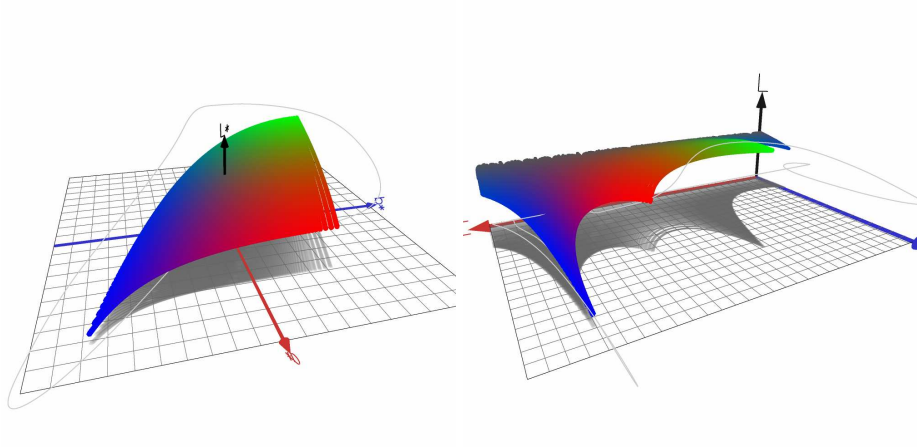
D'autres distances couleurs ont éte introduites, dans des contextes industriels ou par la norme CIE. Elles ont comme avantage de gérer plus finement le phénomène d'adaptation chromatique. Néanmoins, dans le contexte d'indexation d'images où nous nous situons, nous n'avons pas jugé opportun d'explorer ces voies là, ne serait-ce que pour la raison simple de non connaissance des conditions d'acquisitions précises des images initiales.

1.3.1.7 Espaces couleurs : récapitulatif

Le tableau 1.1 résume les principaux espaces couleurs que nous avons utilisés, ainsi que leurs qualités, du moins dans une optique de recherche par le contenu. Il peut sembler, a priori, que la panel utilisé est large mais l'influence de l'espace couleur sélectionné est grande. D'abord l'espace couleur influe sur l'algorithme lui-même, d'un point de vue conception par exemple: discrétiser les couleurs dans l'espace RGB n'a pas le même sens que discrétiser l'espace HSV. Quelque soit l'objectif voulu, il faut toujours rechercher la représentation, ie l'espace couleur, qui sera le mieux adapté aux données et à l'algorithme que l'on souhaite utiliser.

Espace Couleur	Calcul Linéaire	Distance Uniforme	Avantages Inconvénients
RGB, …	Oui	Non	• Format de base • Nombreux algorithmes • Axes fortement corrélés
XYZ, …	Oui	Non	• Espace incontournable • Décomposition Luminance/Chrominance • Nécessite de connaître les conditions d'acquisitions
$X_1X_2X_3$	Oui	Non	• Axes décorrélés • Forte complexité algorithmique • Espace lié à l'image étudiée
$I_1I_2I_3$	Oui	Non	• Approximation de la décorrélation • Calcul beaucoup plus rapide que $X_1X_2X_3$ • Dépend des images sélectionnées pour le calcul de la matrice de passage
HSV, …	Non	Oui	• Séparation Luminance, Teinte et Saturation • Bonne corrélation avec la représentation humaine des couleurs • Extraction de la teinte • Transformation non-linéaire : création d'artéfacts numériques
YIQ …	Oui	Non	• Décomposition Luminance/Chrominance • Meilleure décorrélation que RGB • Très utilisé en Vidéo
$L^*a^*b^*$ …	Non	Oui	• Distance adaptée à la perception humaine • Décomposition Luminance/Chrominance • Temps de calcul important • Transformation non-linéaire : création d'artéfacts numériques • Nécessite de connaître les conditions d'acquisition • Non judicieux si la source RGB est de 8bits ou moins

Tab. 1.1 – *Récapitulatif des principaux espaces couleur*



(a) Espace $L^*a^*b^*$ (b) Espace LHC

Fig. 1.10 – *Triangle de Maxwell dans les espaces* $L^*a^*b^*$ *et* LHC

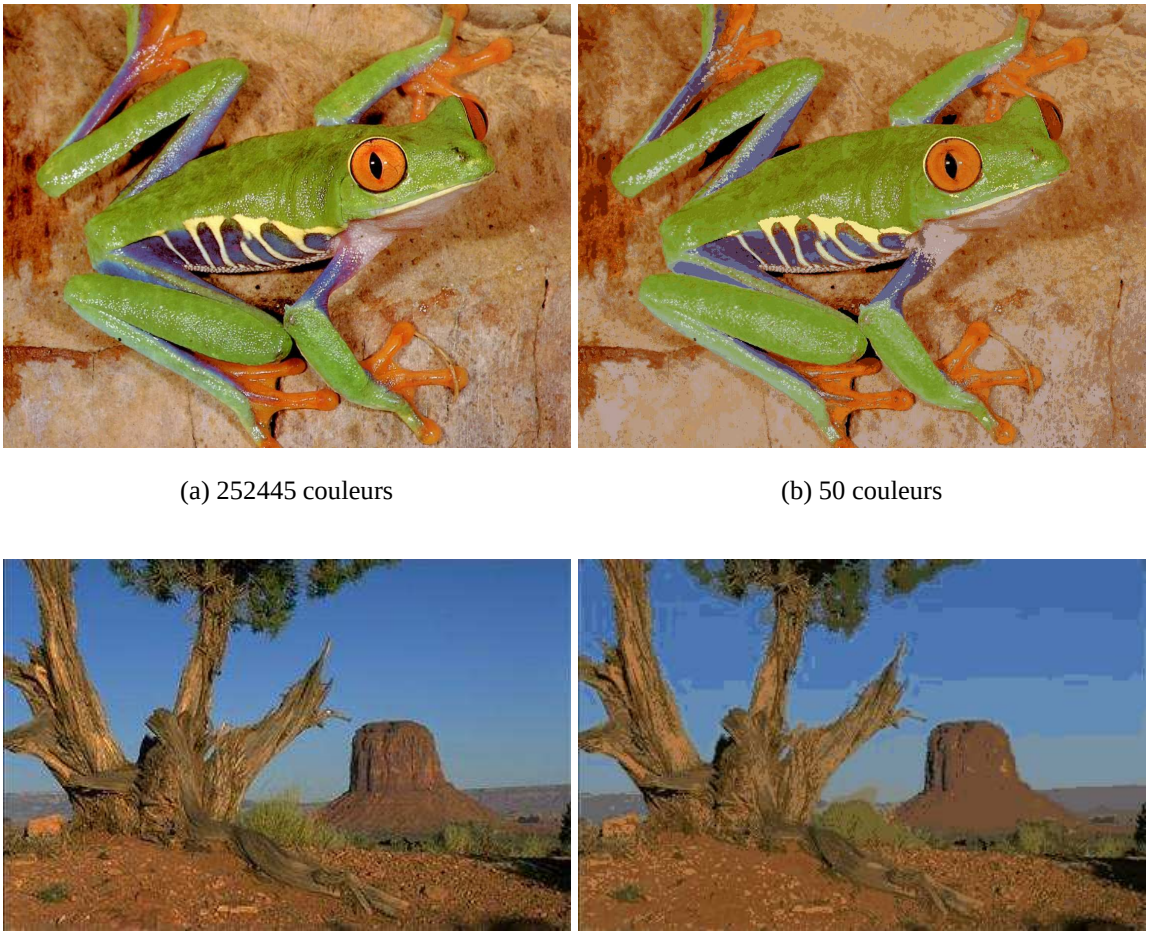
1.3.2 Quantification visuelle

De nombreuses démarches pour l'extraction de signatures, basées sur la couleur, se sont orientées vers une quantification préalable de toute image. Cela permet de réduire l'information couleur de façon considérable sans perte d'informations conséquente (dans le cadre recherche par similarité bien sûr). On parle donc ici de paléttisation, ie la quantification dite visuelle. Comme l'illustre la figure 1.11, réduire plus de 250000 couleurs à 50 couleurs différentes ou plus de 37000 à 20, ne semble pas en effet altérer notre perception générale de l'image : les images doivent être considérées comme similaires. De plus, dans l'objectif d'extraire des signatures couleurs, il est évident que des espaces quantifiés permettent d'obtenir des descripteurs de taille réduite. Rappelons ici que nous entendons par signature couleur tout attribut susceptible de décrire la distribution colorimétrique d'une image.

La quantification, si elle est nécessaire à l'extraction de signatures, est régie par différents procédés :

- Quantification statique
  Le nombre de couleurs voulu est prédéfini, ie quelle que soit l'image, le nombre de couleurs résultantes est le même. Les couleurs peuvent être obtenues via une pré-discrétisation d'un espace couleur ou par un algorithme de clustering où le nombre de germes est fixé a priori.
- Quantification dynamique
  Le nombre de couleurs varie en fonction du contenu de l'image. Cette approche permet en fait d'adapter le nombre de couleurs en fonction de la distribution colorimétrique. Cela est très important dans le cas où l'image est fortement texturée ou si elle est très com-

plexe. En effet, ces méthodes tentent d'assurer une bonne quantification mais sans perte d'information fondamentale, ce qui est primordial pour extraire une signature ensuite.



(a) 252445 couleurs (b) 50 couleurs

(c) 37044 couleurs (d) 20 couleurs

Fig. 1.11 – *Exemples de quantification par l'algorithme du MeanShift*

Néanmoins, si quantifier une image peut être judicieux dans certains cas, automatiser cette étape comme pré filtrage dans le cadre de recherches d'images par le contenu peut conduire à des incohérences. La quantification, en effet, perturbe considérablement le nuage couleur et les contours de l'image, caractéristiques souvent primordiales à l'extraction de certains paramètres. La quantification, même si elle se justifie visuellement, doit être utilisée afin de concentrer l'information couleur uniquement si les descripteurs extraits en aval le permettent.

Notons par contre qu'une quantification est souvent inhérente au format d'image utilisé et donc à l'algorithme de compression associé[Trémeau *et al.*, 2004]. La conséquence de ce choix de stockage sur les descripteurs n'est pas négligeable[Jolion et Bres, 1999].

1.3.3 Statistiques du nuage colorimétrique

Il est possible d'extraire directement des informations à partir du nuage formé des couleurs de chaque pixel. Nous pouvons citer le calcul des différents moments : Soit Ω l'ensemble des pixels considérés et $\Phi(\omega)$ sa coordonnée couleur dans l'espace choisi:

1. Moyenne

$$M_1 = \mu = \frac{1}{\|\Omega\|} \cdot \sum_{\omega \in \Omega} \Phi(\omega) \tag{1.12}$$

2. Écart-type

$$M_2 = \sigma = \sqrt{\frac{1}{\|\Omega\| - 1} \cdot \sum_{\omega \in \Omega} (\Phi(\omega) - \mu)^2} \tag{1.13}$$

Dans le cas d'images de scènes, le calcul des moments sur l'image complète n'est pas justifié. Par contre, dans le cas de régions, cela peut être un très bon descripteur. La distance associée à ces différents moments peut prendre différentes formes, mais la plus utilisée est la classique distance euclidienne dans l'espace RGB.

1.3.4 La recherche par histogramme

Un histogramme peut être considéré comme la modélisation probabiliste d'une image : une approximation de la densité de la variable aléatoire image. Si $[C_1,C_2,...C_n]$ est l'ensemble des n couleurs de l'image représentée, alors l'histogramme est défini par : $[H_1,H_2,...H_n]$ où H_i représente le nombre d'occurrences de la couleur C_i dans l'image. En général, on normalise l'histogramme, ie : $\forall i \in [1..n]\ h_i = \frac{H_i}{\sum_{k=1}^{n} H_k}$. Notons que $\sum_{k=1}^{n} H_k$ correspond au nombre de pixels de l'image.

Première remarque, l'espace couleur sélectionné pour extraire l'histogramme est très important, ainsi que la méthode de quantification utilisée. Les signatures histogrammes sont considérées, à juste titre comme insensibles à de nombreuses transformations, et robustes, dans une certaine proportion, aux effets de compression, de changements d'échelles ou de rotation. Pour rester indépendant de l'illuminant, nous pouvons utiliser uniquement les composantes H et S de l'espace HSV par exemple : la teinte et la saturation étant évidemment indépendantes à la luminance. Une comparaison de différents modèles colorimétriques réalisée par Gevers et Smeulders [Gevers et Smeulders, 1996] donne une fine critique des différentes approches. D'autres modèles [Funt et Finlayson, 1995] sont basés sur les dérivées directionnelles du logarithme des couleurs dans l'espace RGB: l'aspect voisinage permet l'indépendance aux effets de l'illuminant.

Une évolution intéressante dérive d'un domaine bien connu issu de l'intelligence artificielle : l'approche floue [Grecu et Lambert, 2001]. Il s'agit, dans le contexte des histogrammes, de cal-

culer des quantificateurs d'appartenance à telle ou telle partie de la discrétisation. Ainsi la valeur attribuée H_i ne correspond plus au nombre de pixels de couleur C_i mais à la somme, pour chaque pixel, du degré d'appartenance à la classe C_i. Ainsi, nous atténuons l'effet de discrétisation lié aux histogrammes, la décomposition floue est plus "continue".

Quels que soient la représentation ou l'espace couleur sélectionnés pour construire l'histogramme, le problème du choix de la distance entre deux histogrammes se pose. Ce choix va très largement influencer le type de similarité induit par cette distance. Citons ici les principales méthodes représentatives des différentes approches pour évaluer une distance entre histogrammes. Notons que nous parlons de distance souvent par abus, car au sens mathématique du terme, la plupart sont des mesures et non des distances.

- Distances classiques
  La distance euclidienne, ou plus globalement les distances de Minkowski L_k sont très utilisées pour leur simplicité, leur rapidité de calcul et aussi car il s'agit véritablement de distances mathématiques. La distance s'exprime sous la forme :

$$d_{L_k}\left(H^1,H^2\right) = \left(\sum_{i=1}^{n}\left(H_i^1 - H_i^2\right)^k\right)^{\frac{1}{k}} \tag{1.14}$$

  Plus précisément pour L_2 et L_∞

$$d_2\left(H^1,H^2\right) = \sqrt{\sum_{i=1}^{n}\left(H_i^1 - H_i^2\right)^2} \tag{1.15}$$

$$d_\infty\left(H^1,H^2\right) = \max_{1\leq i\leq n}\left\|H_i^1 - H_i^2\right\| \tag{1.16}$$

  L_2 est la distance Euclidienne. L_∞ mesure l'ecart maximal entre deux classes.
  Il s'agit donc de distances classe à classe (figure 1.12) ou bins à bins, qui ont le principal défaut de comparer statiquement chaque classe des histogrammes sans tenir compte du voisinage. Les figures 1.12 et 1.13 illustrent cette comparaison classe à classe et le principal défaut engendré. En effet, alors que deux histogrammes sont à égale distance d'un histogramme référence il semble clair que l'un est plus proche qu'un autre. Ce problème, qui finalement se retrouve dans de nombreux contextes, pas seulement dans le cas d'histogrammes, signifie qu'on ne peut pas mesurer une similarité sans prendre en compte le voisinage proche de l'histogramme.
  D'autres distances dérivant des distances de Minkoswki ont été introduites dans le contexte de recherche de similarité : la distance de Geman-McClure [Geman et McClure, 1987], la distance pondérée de Huang [Huang *et al.*, 1998].

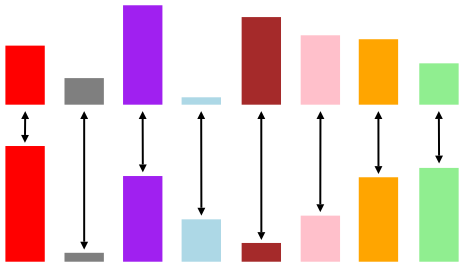


Fig. 1.12 – *Histogramme : distance bins à bins*

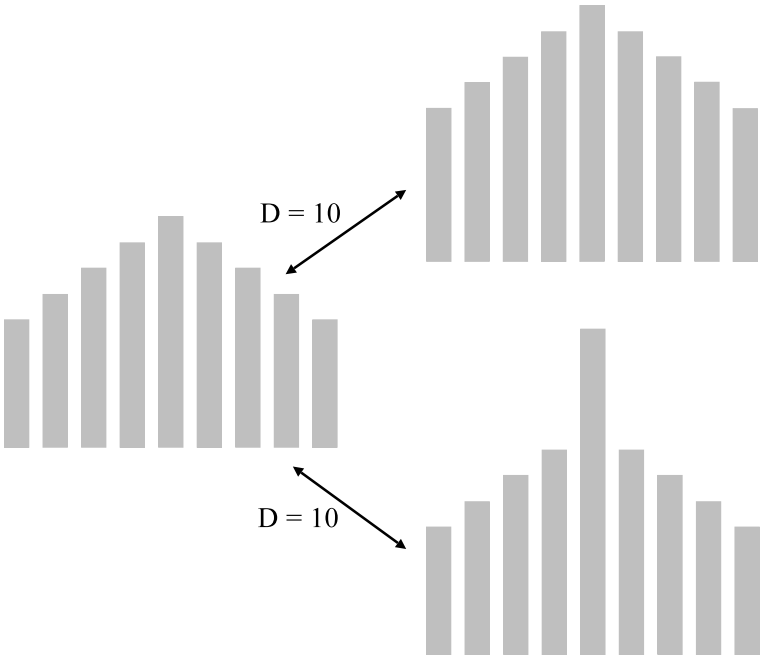


Fig. 1.13 – *Défaut d'une distance bins à bins*

- Distance par intersection

  Swain et Ballard [Swain et Ballard, 1991] proposent une mesure basée sur l'intersection d'histogrammes. Cela permet en fait d'évaluer le recouvrement de deux histogrammes. Bien sûr, le recouvrement se fait sur un histogramme normalisé. La distance de Swain et Ballard s'exprime ainsi :

$$d\left(H^1,H^2\right) = 1 - \frac{\sum_{j=1}^{n} \min\left(H_j^1,H_j^2\right)}{\sum_{j=1}^{n} H_j^1} \tag{1.17}$$

  Cette distance n'en est pas une: elle ne respecte pas la propriété de symétrie, à cause du dénominateur.

- Distances issues de la distribution

  Le principe ici est de considérer les histogrammes comme la densité de probabilité de la variable aléatoire image. Ainsi, on peut appliquer divers tests d'hypothèse suivant telle ou telle loi de probabilité. Il s'agit, par exemple, à partir de deux histogrammes, modélisant deux distributions, de calculer la probabilité que ces deux distributions soient issues de la même loi.

  Une des lois très utilisée est celle du χ^2, pour laquelle Schiele [Schiele et Crowley, 1996], dans une comparaison de distances entre histogrammes, conclut qu'elle donne les meilleurs résultats. Rappelons la définition du test du χ^2, où l'on considère des distributions gaussiennes.

$$d_{\chi^2}\left(H^1,H^2\right) = \sum_{j=1}^{n} \frac{\left(H_j^1 - H_j^2\right)^2}{\left(H_j^1 + H_j^2\right)^2} \tag{1.18}$$

- Distances pondérées

  Niblack and al [Niblack *et al.*, 1993] ont poposé une distance euclidienne pondérée basé sur une évolution de l'espace Munsell. Le principe général est de considérer les 4096 couleurs initiales de l'espace de Munsell pour réaliser une étape de clustering, après une quantification dans RGB en 256 couleurs. Les couleurs principales de chaque image sont extraites et permettent de calculer le dit histogramme. La distance évalue ainsi la corrélation croisée entre les deux histogrammes, pondérée par la distance entre chaque couleur dans l'espace de Munsell. Ainsi, la quantification utilisée n'est pas statique sur l'ensemble des images mais adaptée pour chaque image.

- Distance perceptuelle

  Rubner [Rubner, 1999] a proposé en 1999 une adaptatation pour l'indexation d'images de la distance EMD, Earth Mover Distance. Sans supposer aucune propriété sur les histogrammes ou la quantification utilisée, cette distance entre histogrammes évalue la quantité d'energie nécessaire pour transformer une distribution H^1 en une distribution H^2. Il s'agit donc de trouver le chemin minimal pour passer du premier histogramme au second,

par échange d'énergie. Cette métrique constitue une distance perceptuelle de comparaison d'histogrammes. Ainsi son utilisation dans le contexte de recherche d'images par le contenu est judicieux. Néanmoins, cette distance souffre d'une phase de résolution sous contrainte de type Simplex. Or cette classe d'algorithmes induit un fort coût algorithmique.

1.3.5 Vecteur de cohérence couleur

Extension des méthodes par histogrammes, la cohérence d'un pixel définie par Pass en 1996 [Pass *et al.*, 1996] est le degré d'appartenance de la couleur à de larges régions spatiales couleur homogènes. Pour chaque couleur de l'histogramme, un vecteur de cohérence est alors défini par le nombre de pixels cohérents et par le nombre de pixels incohérents. Ainsi, au lieu d'obtenir un histogramme de type *(Couleur,Nombre de pixels)* on obtient un histogramme de type *(Couleur, Nombre de pixels cohérents, Nombre de pixels incohérents)*. La distinction entre ces deux classes de pixels permet d'achever plus finement la phase de recherche de similarité. Cette phase est similaire ensuite aux approches par histogrammes.

1.4 Description de la texture

La texture est une composante visuelle intimement liée à la couleur. L'information texture permet à l'oeil humain de reconnaître un aspect, une surface et ainsi de recomposer le contenu de la zone texturée [Marr, 1982]. Détaillons brièvement différentes approches permettant de quantifier les différents aspects d'une texture.

1.4.1 Différents types de textures

Le monde de l'image a tendance à séparer la texture en deux grandes parties. D'un coté des textures "déterministes", c'est à dire des images où nous pouvons extraire des primitives et des lois de placement déterministes. À l'opposé, les approches stochastiques qui s'orientent vers une vision purement aléatoire des primitives composant l'image et de leur loi de placement. Par contre, si nous demandons à un utilisateur de définir les notions lui permettant de caractériser une texture, celui ci utilisera un certain nombre de sensations visuelles. Le tableau 1.2, issu de [Asendorf et Hermes, 1996] récapitule les paramètres visuels que les différents auteurs ont retenus [5].

La problèmatique consiste à extraire des paramètres susceptibles de quantifier chacune de ces propriétés visuelles.

5. Tam. [Tamura *et al.*, 1978], Ama. [Amadasum et King, 1989], Rao. [Rao et Lohse, 1989], Wu. [Wu et Chen, 1992] et Ase. [Asendorf et Hermes, 1996]

Property	Tam.	Ama.	Rao.	Wu.	Ase.
linelikeness	√				√
bloblikeness	√				√
multiareas					√
planarity					√
coarseness	√	√		√	√
directionality	√		√		√
regularity	√		√	√	√
contrast	√	√		√	√
roughness	√			√	
complexity		√	√		
periodicity				√	
busyness		√			
texture strength		√			
softness					√

Tab. 1.2 – *Différents descripteurs visuels de textures*

1.4.2 Signatures de texture

Nous entendons par signature divers descripteurs qui indiquent une certaine caractéristique de la texture. Citons ici des descripteurs comme la direction, le contraste... Il s'agit d'attributs qui se veulent "perceptuels". Nous retrouvons une grande majorité de ces descripteurs dans [Rao, 1990].

1.4.3 Analyse par répétabilité locale

Étudier dans un voisinage raisonnablement important les différentes répétitions et discontinuités autour de chaque pixel doit permettre ainsi de mesurer certains critères d'une texture. Nous avons retenu deux approches similaires, d'une part les matrices de co-occurrences et d'autre part les matrices de longueur de plages.

Introduites en 1979 [Haralick, 1979] sous le nom de SGLDM (*S*patial Grey Level Dependance Method), les matrices de co-occurrence forment un outil très intéressant dans l'analyse de textures. Cette matrice est sensée modéliser les relations d'intensité entre les différents pixels. Ainsi la texture se caractérise à l'aide de paramètres localisant l'information dans cette matrice.

Soit une image définie par la fonction d'intensité Φ, t un vecteur du plan; $[h_t]$ est la matrice

de co-occurrence définie par :

$$h_t(i,j) = \#\left\{(u,u+t) \in I^2; \Phi(u) = i \wedge \Phi(u+t) = j\right\} \tag{1.19}$$

Il est ensuite possible de faire varier le vecteur t de telle façon à privilégier soit une distance, soit une orientation.

Des paramètres statistiques classiques sont extraits en définitive sur cette matrice, comme l'inertie, l'entropie, l'homogénéité ou encore la corrélation.

L'idée principale des matrices de longueur de plages est d'utiliser une information de type "Run length", ie longueur de pixels de même couleur. Plusieurs auteurs ont introduit cette approche dont notamment Galloway [Galloway, 1974] pour les textures en niveaux de gris. Décrivons succinctement une extension couleur [Vertan *et al.*, 2002b].

Le principe d'une longueur de plage est de compter le nombre de pixels de même couleur C dans une direction θ. Cela permet ainsi de construire la matrice de longueur de plages couleur, sur laquelle les auteurs proposent, d'extraire différents paramètres dont, entre autres :

- Nombre de plages couleurs
- Hétérogénéité d'une couleur ou d'une longueur
- Proportion de longues plages

Il faut noter qu'une étape de quantification est nécessaire, réduisant la complexité colorimétrique sans altérer la texture. Ainsi des couleurs sont similaires (pour la calcul de la longueur de plage) si elles sont de la même palette colorimétrique ou bien si la distance (les auteurs préconisant une distance dans $L^*a^*b^*$) entre les deux est non différentiable ou presque.

1.4.4 Modèles fréquentiels

Les transformations en fréquences sont toujours d'actualité pour la reconnaissance de textures. En effet, elles sont très bien adaptées pour détecter certains phénomènes bien précis des textures. On pense notamment aux textures multi-échelles, où une hiérarchie de motifs constitue la texture.

Les modèles classiques basés sur des transformées de Fourier ont montré leurs efficacités dans de nombreuses applications. Cependant, dans le cadre des modèles fréquentiels, ils semblent être supplantés par les approches filtres de Gabor ou ondelettes. Les bancs de filtres de Gabor sont particulièrement adaptés pour discriminer les textures. Un filtre de Gabor peut être vu comme un filtre passe bande avec une enveloppe gaussienne. Différentes orientations et tailles peuvent être alors données aux filtres, ce qui permet de constituer pour une application donnée un banc de filtres. Sur les images sous-bandes résultantes, le calcul des moyennes, variances ou autres statistiques simples, permet d'effectuer par la suite la reconnaissance des textures [Jain et Healy, 1998].

On peut ajouter aux modèles fréquentiels la “Steerable Pyramid”. Il s’agit d’une structure multi-échelle pyramidale orientée. En effet, le passage d’un niveau à l’autre se fait via un filtre orienté permettant ainsi de privilégier une direction pour une sous bande. Les statistiques issues de cette pyramide permettent ensuite l’identification de textures [Blancho et Konik, 2000, Blancho *et al.*, 1998].

1.5 Décrire la forme

La forme est généralement une description très riche d’un objet. En effet, certaines formes sont fortement caractéristiques de l’objet qu’elles représentent; on peut penser à la forme d’une pomme ou d’un avion. Il semble donc naturel de vouloir décrire un objet par sa forme, si tant est que nous soyons capables d’extraire parfaitement l’objet. Dans ce cas, c’est-à-dire si nous possédons une image binarisée (0 le fond, 1 l’objet), il est possible d’extraire des descripteurs efficaces [Zhang et Lu, 2001], ou bien de mettre en place différentes métriques pour évaluer la similarité de deux formes. Trois importantes approches co-existent : les méthodes ayant pour but l’extraction de paramètres numériques, celles qui comparent deux formes par transformations géométriques et finalement les méthodes hybrides. Ces dernières décrivent numériquement un certain nombre de points ou régions émergents de la forme, permettant ensuite de comparer géométriquement la position des points ou des régions. Sans vouloir être exhaustifs, parcourons ces différentes approches.

1.5.1 D’une forme à un vecteur numérique

Distinguons tout d’abord deux types d’approches pour extraire un vecteur numérique d’une forme: soit la description de la région incluse dans la forme, soit la description du contour de la forme. Dans le cas de la description de la région, citons quelques paramètres classiques en traitement d’images :

- Aire
- Périmètre
- Élongation - Rapport entre la plus grande corde et la perpendiculaire associée.
- Compacité - Rapport entre le carré du périmètre et l’aire.
- Primitives géométriques englobantes (la plus petite englobante) et incluses (la plus grande incluse). Le descripteur est alors un ratio entre l’aire ou le périmètre de la région et les paramètres de la primitive (qui peut être un cercle ou un rectangle par exemple). La circularité, ie le rapport entre le périmètre et celui du cercle englobant est très souvent utilisée en traitement d’images pour l’analyse des formes.

- Nombre d'Euler - Le Nombre d'Euler est un entier associé à toute surface orientable. C'est un invariant topologique de celle-ci, dans le sens qu'il ne change pas si la surface subit une déformation continue. Dans le cas des formes géométriques du plan, il s'agit du nombre de composantes connexes de la figure moins le nombre de trous.

Un autre descripteur global de forme est basé sur le calcul des différents moments [Hu, 1962] associés à la région. QBIC [Flickner *et al.*, 1995], par exemple, utilise des invariants sur les moments centrés normalisés de la région. L'intérêt de ces derniers est d'être invariants pour de nombreuses opérations géométriques comme la rotation, la translation ou les changements d'échelle. Par contre, cette approche est très sensible au bruit et aux déformations, même minimes. Pour rappel, les moments d'ordres k,l sont définis comme suit :

$$\sum_{x,y} x^k y^l f\,(x,y) \tag{1.20}$$

L'ensemble infini $k = 0,1,2,....; l = 0,1,2,....$ définit de manière unique toute forme géométrique de l'espace discrétisé.

La transformation ART, Angular Radial Transform [Agnihotri, 1999], forme un des deux descripteurs de formes de la norme MPEG-7. Grossièrement il s'agit de modéliser la direction et le rayon de la courbe.

Remarquons qu'il est aussi possible d'utiliser les coefficients de la transformée de Fourier discrète ou ceux de la décomposition en ondelettes. Ainsi la distance est évaluée dans le domaine fréquentiel.

1.5.2 D'une forme à une autre

Une autre approche de la reconnaissance de formes consite à calculer la distance géométrique entre une forme et une autre. Il peut s'agir de l'énergie pour faire évoluer une forme vers celle de référence, ou bien encore, de chercher un jeu de transformations géométriques permettant de passer d'une forme à une autre.

La forme est soit comparée à un ensemble de formes pré-établies, soit à une forme de référence. Dans le premier cas, cet ensemble est constitué de peu de formes, dont nous connaissons a priori le contenu : une série de formes de pomme, de poire ou de banane, si on se place dans la reconnaissance de fruits par exemple. Ainsi il s'agit de faire de la correspondance entre les différents objets et l'ensemble référence. Il est donc dans ce contexte précis, possible de répondre à des questions du type : "Présence ou non d'un cheval sur la photographie ?".

Différentes méthodes peuvent être employées, citons les plus importantes :

- Jeu de transformations - déformations

À partir d'un jeu de transformation, il s'agit de calculer le jeu minimal (en terme de coûts) pour transformer une forme initiale en une autre [Bimbo *et al.*, 1994, Kass *et al.*, 1998].

- Distance entre courbes
  Le calcul de la distance de Hausdorff [Huttenlocher *et al.*, 1993] ou encore celle de Fréchet [Alt et Godau, 1995] permet de quantifier la distance entre deux courbes, et ainsi entre deux formes.

On peut signaler que ces méthodes ne permettent pas d'indexer à proprement parler une base d'images (dans le sens créer un index) mais induisent une comparaison entre l'objet recherché et toutes les images de la base.

1.5.3 D'une forme à un ensemble de vecteurs numériques

L'idée principale de ce type de descripteurs est de modéliser la courbe par un ensemble de points caractéristiques et par les relations qui existent entre ces derniers. Cette approche a permis le développement de nombreuses méthodes. Voyons succinctement les plus usitées:

- CSS - Curvature Scale Space
  La représentation d'une courbe dans cet espace permet par exemple d'obtenir une série de points de courbure qui vont permettre la recherche de similarité avec d'autres formes [Mokhtarian *et al.*, 1996]. Signalons que la norme MPEG-7 [Agnihotri, 1999] utilise CSS en approche contour.
- Codage du contour
  Évidemment, le codage de Freeman [Freeman, 1974] et les approches similaires de la géométrie discrète [Montanvert et Chassery, 1993] permettent de coder l'information contour. Il est ainsi possible de reconstituer la forme à partir de ce codage.
- Points de courbures
  Une autre possibilité est de détecter un certain nombre de points clés de la courbe. La similarité est ainsi obtenue en comparant les différentes distributions spatiales des points, que l'on peut précédemment apparier via leurs caractéristiques locales. En effet, chaque point peut être décrit par ses coordonnées, mais aussi par le rayon de courbure, la direction ou d'autres données locales géométriques.

1.6 Quel descripteur sur quelle donnée ?

Nous avons vu une petite partie de l'importante batterie de descripteurs possibles, tant pour la texture, la couleur que la forme. Le problème majeur est certes de coupler tous ces paramètres mais aussi de faire un choix stratégique : sur quelles données doit-on les calculer ?

1.6.1 L'approche globale

L'image est vue dans sa globalité, comme un tout non séparable. Ainsi, les descripteurs sont calculés sur l'ensemble de l'image, sans discernement. Le principal avantage est de traiter l'image globalement, d'où l'invariance à de nombreuses transformations. Il s'agit aussi généralement de paramètres facilement extractibles, et, il n'y a qu'une seule donnée, de faible taille pour le stockage. Les meilleurs exemples d'approche globale sont sans doute les mesures sur histogrammes.

Par contre, l'approche globale, de par sa définition même, peut aboutir à deux valeurs identiques pour deux images très différentes. En effet, globaliser une information locale engendre un mixage d'informations disparates et une perte de l'information locale. Or cette information locale, où se détachent des zones et des relations entre celles-ci, se retrouve atrophiée dans une approche globale. De plus, visuellement, si le cerveau humain analyse un image globalement, il analyse aussi, voir principalement, l'image en zones [Marr, 1982]. Ainsi, étudier une image en décrivant les zones qui la composent est incontournable dans une optique de recherche de similarités.

1.6.2 L'approche points d'intérêt

Les points d'intérêt sont dérivés, à l'origine, d'une volonté de caractériser les zones contenant le plus d'information visuelle. Néanmoins, les différentes méthodes sont souvent éloignées de cela, dans le sens où elles recherchent plutôt à être insensibles à certaines transformations géométriques. Le choix de sélectionner les coins est le cas le plus répandu. La qualité principale de ces méthodes est d'être relativement stable lors des transformations classiques comme la rotation, la translation, le sur ou sous éclairement ou encore les effets d'échelle. Ces propriétés amènent une certaine robustesse lors d'une approche indexation.

En pratique, trois classes de méthodes d'extraction de points d'intérêt interviennent.

- Méthodes différentielles

  Cette classe de méthodes utilise des invariants différentiels pour extraire les points d'intérêt d'une image. La plupart sont des adaptations d'un détecteur de coin, le détecteur dit de "Harris"[Harris et Stephens, 1988]. Les adaptations sont souvent liées à la reconnaissance de formes, par exemple par appariement des points [Schmid, 1996].
- Méthodes multi-échelles

  L'utilisation de pyramides de contraste[Jolion et Bres, 1999] ou de transformations en ondelettes [Loupias *et al.*, 2000] permet d'extraire des points qui, s'ils ne sont plus forcément des coins, renforcent l'information visuelle portée par eux. Ce sont des méthodes particulièrement robustes face à la compression et au bruit.

- Méthodes par “filtrage”
  Ces méthodes ont pour avantage de bien réagir en présence de bruit important, en utilisant des filtres circulaires adaptatifs comme la méthode SUSAN [Smith et Brady, 1997] ou des filtres directionnels.

Fig. 1.14 – *Exemple de points d'intérêt, méthode de SUSAN*

Les points détectés comme illustré) la figure 1.14, correspondent le plus souvent à des coins de l'image. Ces derniers peuvent alors servir dans un objectif de recherche de similarité, par deux approches distinctes :

- Invariants locaux
  En calculant des invariant locaux autour de chaque point d'intérêt détectés, il est possible ainsi d'établir une certaine similarité entre images. Citons le moteur KIWI (Key-points Indexing Web Interface[6]) qui utilise exclusivement des paramètres issus d'une détection de points d'intérêt.

6. http://telesun.insa-lyon.fr/kiwi/

- Matching d'objet
  Il s'agit sans doute de l'application la plus importante. Il est possible d'établir une mesure de similarité entre images via les points détectés[Sand et Teller, 2004, Schmid, 1996]. La technique généralement utilisée est de retrouver un ensemble de points dans une autre image, notamment grâce aux relations angulaires entre ceux-ci. La reconnaissance qui en découle est multi-échelle et assez stable à de petites variations, tant que les angles et les positions relatives ne varient pas trop. Néanmoins ce type de matching requiert un temps de calculs excessif qui exclue toute utilisation sur de grandes bases d'images.

D'autres usages des points d'intérêt existent, comment pour réaliser du matching en Stéréo Vision[Pollefeys *et al.*, 1998] ou du suivi dans les séquences vidéo[Tissainayagam et Suter, 2005].

De plus, l'approche point d'intérêt se rapporte usuellement à l'indexation d'images de scènes. Néanmoins, ceux-ci peuvent aussi apporter un pouvoir discriminant important en indexation de textures[Da Rugna et Konik, 2001, Da Rugna et Konik, 2002b], singeant de la sorte une approche simpliste de la texture similaire visant à utiliser les maxima locaux comme points clés d'une texture[Mitchell *et al.*, 1977, Karu *et al.*, 1996].

1.6.3 L'approche région

Dans ce contexte, une étape de segmentation permet un pavage de l'image en différentes zones ou régions. Ces dernières sont détectées suivant des propriétés diverses. Tel algorithme privilégie l'homogénéité couleur, tel autre rassemble les zones par textures ou se propose d'extraire de grandes zones d'influences de l'image. Sans discuter ici de cette problématique, il est évident que le choix de l'algorithme de segmentation influence directement le choix des paramètres à associer. Plus en aval de cette approche, d'autres questions se posent, et plus précisément celle de savoir comment comparer deux ensembles de régions, chacun représentant une image. Examinons brièvement les deux démarches communément utilisées:

- L'appariement direct
  Le but est d'apparier par un matching, de type graphe par exemple, les régions de chaque image deux à deux. Ce matching porte ici sur des descripteurs numériques de chaque région, sans inclure de notion spatiale [Ardizzoni *et al.*, 1999, Lew, 2001]. Ainsi deux images A et B sont décrites par A_{F_i},A_{C_i},A_{T_i} et B_{F_i},B_{C_i},B_{T_i}, $1 \leq i \leq N$. Les vecteurs A_{F_i}, A_{C_i} et A_{T_i} décrivant l'information Forme, Couleur et Texture, le calcul de la distance entre deux régions peut être réalisé comme suit:

$$\begin{aligned} D_{ij} \;=\; D\left(A_i,B_j\right) \;&=\; W_F.\textstyle\sum_{k=1}^{f} w_F^k.d\left(A_{F_i}^k,B_{F_j}^k\right) \\ &+\; W_C.\textstyle\sum_{k=1}^{c} w_C^k.d\left(A_{C_i}^k,B_{C_j}^k\right) \\ &+\; W_T.\textstyle\sum_{k=1}^{t} w_T^k.d\left(A_{T_i}^k,B_{T_j}^k\right) \end{aligned} \tag{1.21}$$

W_X est un facteur permettant de pondérer les influences de la Forme, de la Texture et de la Couleur dans le calcul de la distance finale.

- L'agencement spatial
  En plus de l'information couleur, texture ou forme, on essaie de modéliser l'information spatiale dans l'image requête afin de la reconnaître dans l'image comparée. Sans parler de modèles complexes [Bimbo, 1999] comme les modèles hiérarchiques[Dombre, 2003] ou multi échelles, on peut penser aux modèles simples où l'on décrit des relation de types "en dessous" ou "disjoint" par exemple, comme le montrent les figures 1.15 et 1.16. On pense notament ici à des modèles de graphes topologiques Le but premier de ce type d'approche est de s'approcher de l'information sémantique. On peut imaginer décrire un cheval comme un ensemble de régions de couleurs marrons agencées bien précisément. Néanmoins ces méthodes restent insuffisantes, et sont principalement incapables de répondre à la complexité et la diversité du monde réel. Décrire un objet comme un ensemble de régions avec un agencement précis oblige soit à décrire tous les objets existants, ce qui n'est pas imaginable, soit à retrouver cet agencement à partir d'exemples. Et, malheureusement, sur ce dernier point, nous sommes loins des attentes de l'utilisateur. De fait, même si beaucoup plus haut niveau d'un point de vue modèle sémantique, il est difficile d'affirmer que ces méthodes répondent correctement au problème général de l'indexation d'images de scènes.

(a) disjoint (b) superposé (c) recouvert

Fig. 1.15 – *Exemples de relations entre regions*

1.7 Systèmes de recherche d'images par le contenu

Nous avons énuméré, sommairement, les différentes facettes de la recherche d'images par le contenu. Néanmoins, afin d'assembler ce puzzle, arrêtons nous sur la description des systèmes de recherche par le contenu.

(a) (b) (c)

Fig. 1.16 – *Des images avec le modèle: X en dessous de Y*

1.7.1 Principe général

Comme illustré sur la figure 1.17, un système de recherche d'images par le contenu peut se diviser en différentes parties :

- Interface utilisateur "En ligne"
- Le moteur de recherche proprement dit
- La base de données images
- La base de paramètres

1.7.1.1 La base de données images

Le base de données images peut être très diverse. On peut penser à sa taille qui peut être relativement petite, ie quelques centaines d'images, mais aussi beaucoup plus importante, ie plusieurs millions d'images ou plus. Le domaine des images est aussi un point essentiel. Connaître le contenu de la base, ne serait-ce que pour savoir si elle contient ce que l'on cherche, est en soi primordial. C'est l'étendue de ce domaine, tant en taille qu'en diversité, qui prévaut pour le choix des autres éléments d'un CBIR[Bimbo, 1999].

1.7.1.2 La base de paramètres

C'est une base qui est obtenue "hors ligne", le calcul ne dépendant pas de l'utilisateur. Cela permet de stocker un grand nombre de paramètres, parfois complexes à extraire. Ce sont des calculs qui peuvent être très longs, jusqu'à plusieurs jours ou semaines par exemple, si la base d'images est relativement grande. L'ajout d'une image dans la base implique également le calcul de tous les descripteurs sur cette image, et au stockage des résultats dans la base de paramètres. Notons que cette dernière peut aussi stocker un ensemble de mots clés extraits visuellement par des experts.

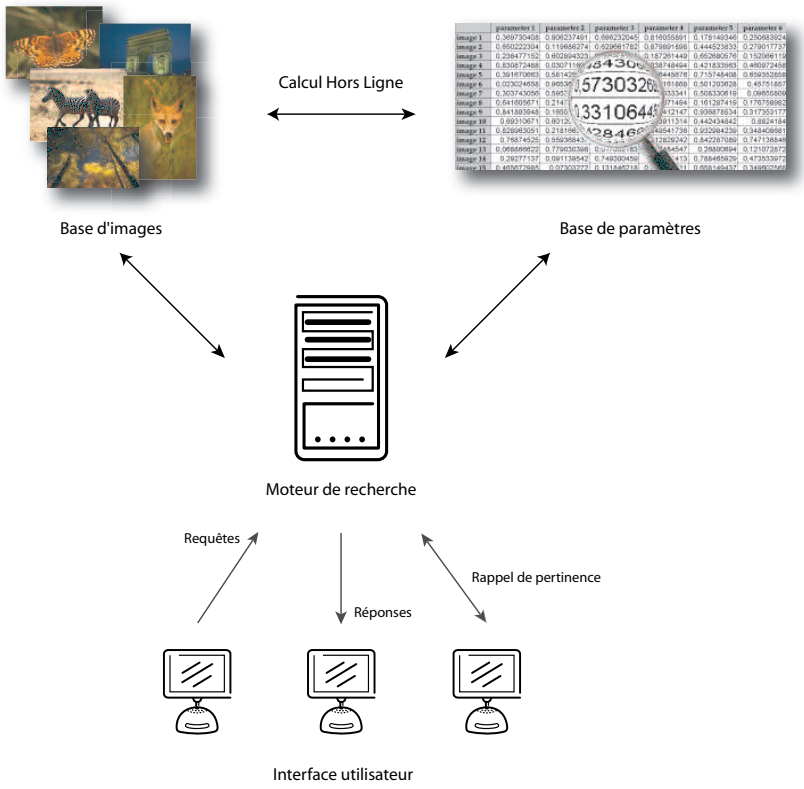


Fig. 1.17 – *Systèmes de recherche d'images par le contenu*

1.7.1.3 Interface utilisateur

L'utilisateur, non expert a priori, a deux attentes :

- Poser sa question - Il arrive avec une idée précise de l'image recherchée, l'interface doit être le lien entre son idée et le moteur de recherche.
- Recevoir les résultats - L'utilisateur doit être tout d'abord satisfait, mais aussi capable d'exploiter les résultats fournis : la forme de la réponse est aussi très importante.

L'utilisateur peut aussi fournir un jugement sur les réponses données, Correct/Non correct par exemple. Cette information peut alors servir au moteur de recherche pour affiner la recherche en cours ou pour améliorer les recherches suivantes.

1.7.1.4 Moteur de recherche

Il centralise toute l'information et effectue les recherches. Pour chacune d'entre elles, le moteur, suivant le passé de l'utilisateur, celui de l'ensemble des utilisateurs et le type de la requête, transforme la recherche en une requête SQL (ou tout autre langage ensembliste) qui va permettre de sélectionner les images réponses.

Explorons maintenant différents points des systèmes de recherche par le contenu.

1.7.2 Les requêtes

La demande de l'utilisateur peut intervernir selon différents types de requêtes. Ce choix influence bien sûr l'interface même dediée à l'utilisateur mais influence surtout la recherche des images escomptées. Voyons tout d'abord les grandes directions proposées habituellement, illustrées par la figure 1.18.

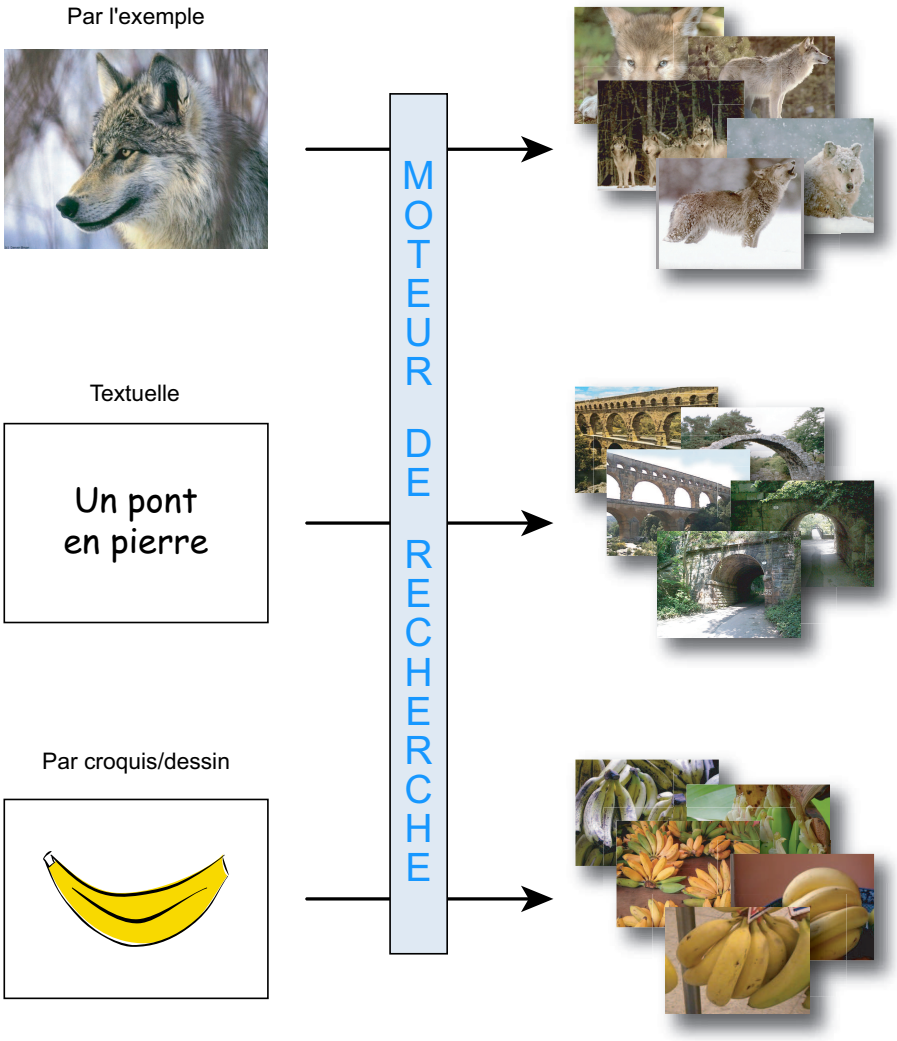


Fig. 1.18 – *Différents types de requêtes*

1.7.2.1 Recherche par l'exemple

C'est l'interface la plus usitée : l'utilisateur propose une image que l'on appelle image requête ou source. Le but du système est ensuite de rechercher des images similaires à l'image requête. Cette dernière peut provenir soit d'une liste existante soit d'une collection propre à l'utilisateur. La notion de similarité ici prend une forme sémantique, ie l'utilisateur attend des images différentes mais ayant un sens similaire à l'image requête.

D'un abord très simple, la requête par l'exemple est liée à un grand nombre de questions sous-jacentes. Quand un utilisateur clique sur une image pour en rechercher les similaires, on se

doit de se poser une question primordiale: quel est le sens de ce clique ? Sur ce point intervient notamment la notion du contexte, lié à l'utilisateur, ou bien encore de l'idée préconçue que se fait ce dernier de l'image recherchée. De plus, quand ce clique est déclenché, l'utilisateur veut-il des images similaires à toute la scène ou seulement à une partie ? Si la figure 1.19 était une image requête, que chercherait l'utilisateur ? Diverses réponses co-existent : un avion, un avion en phase d'attérissage, deux avions superposés ou encore un avion attérissant le matin au bord de l'eau ! La recherche par l'exemple, même sur une image a priori "simple" ou "pauvre", ne sera elle jamais simple.

Fig. 1.19 – *Recherche par l'exemple*

1.7.2.2 Recherche d'objets

Très similaire, dans un certain sens à la recherche par l'exemple, le but ici est de rechercher un objet précis dans une série d'images [Neumann *et al.*, 2002]. Le terme objet est pris ici au sens large, à savoir qu'il peut prendre la forme d'un logo, d'un visage, d'une voiture... On peut rechercher par exemple toutes les images où apparaîtrait le logo d'une certaine marque. On désire ainsi retrouver une partie où toutes les images avec un objet bien défini et écarter toutes les autres. Il s'agit donc d'une similarité non plus sémantique mais exacte.

1.7.2.3 Recherche par croquis - dessin

Généralement l'utilisateur est confronté à une impasse dans les cas précédents : il a une bonne idée de ce qu'il cherche mais pas d'exemples à fournir. Un solution naturelle est de lui faire dessiner une ébauche qui va ainsi servir "d'exemple" à la recherche.

- Le croquis [Egenhofer, 1997] - L'utilisateur décrit ce qu'il désire en représentant précisément les contours des objets, généralement en un seule et unique couleur : seul l'aspect forme porte l'information.
- La dessin grossier [Sciascio *et al.*, 1999] - Un dessin coloré est proposé par l'utilisateur : il contient une représentation colorée de chaque objet, mais où les contours sont généralement vagues. L'information couleur (les couleurs mêmes mais aussi l'agencement de celles-ci) est donc primordiale, au contraire de la forme qui n'est pas très représentative.

1.7.2.4 Recherche par requête textuelle

Le graal de la recherche d'image par le contenu. . . . En effet, une personne, cherchant une image, ne veut ou ne peut ni donner une image exemple, ni même dessiner quelque chose d'approchant. Elle veut juste des images répondant à ce qu'elle cherche : une poignée de main entre deux présidents pour un journaliste, une photo de sa fille à deux ans pour une mère mélancolique. L'utilisateur écrit donc sa requête : le rôle du sytème est de convertir un texte en stratégie de recherche d'images. Nous sommes vraiment ici à l'intersection de l'intelligence artificielle et du traitement d'images.

Les différentes méthodes ne sont pas forcément exclusives. On peut ainsi imaginer une prérecherche via un croquis qui serait ensuite affiné par une recherche par l'exemple. Parallèlement, la stratégie de recherche issue de la requête, s'adapte évidemment au type même de la requête.

1.7.3 Quel modèle d'intégration des descripteurs ?

La gestion des connaissances[Bachimont, 2003] accumulées sur l'image n'est pas une gestion aisée. En effet, comment combiner par exemple un vecteur numérique, donnant une information couleur globale sur l'image et un vecteur numérique calculé à partir de la forme des régions segmentées ? Diverses approches sont possibles:

- Itératives
  Chaque descripteur permet de façon isolé de retrouver les images les plus proches. Ainsi, pour chaque descripteur, on calcule la distance de cette image aux images de la base. On obtient ainsi pour chaque image une position relative à chaque descripteur. La cooptation de tous les descripteurs se fait en élisant la méthode la plus proche. Plus précisément l'image dont la somme des positions est la plus faible sera considérée comme la plus proche. Cette approche permet un traitement très rapide. Néanmoins le fait qu'un descripteur puisse être plus ou moins discriminant qu'un autre n'est pas pris en compte.
- Pondérées
  La distance entre deux images est calculée comme étant la somme pondérée de distances entre les descripteurs. Chaque descripteur est auparavant normalisé. Les poids peuvent être tous égaux mais aussi définis de sorte qu'un descripteur, qui serait plus discriminant, soit avantagé. Ainsi les descripteurs sont intégrés en s'adaptant avec le domaine des images rencontré.
  Les distances pondérées à poids fixes peuvent avoir un défaut, celui de ne pas évoluer similairement aux requêtes et aux images. En fait, tout en s'appuyant sur un modèle pondéré, il semble logique par exemple que les poids évoluent en fonction de la base d'images utilisée (coopération par apprentissage) ou bien encore en fonction de l'utilisateur du sytème. Dans ce dernier cas, il s'agit d'un contrôle de pertinence que nous détaillerons au

paragraphe suivant.

- Pré-filtrage
  Pré-filtrer une base d'images correspond à éliminer les images dont on suppose qu'elle ne peuvent être solutions à la requête[Castelli *et al.*, 1998, Hafner *et al.*, 1995] . Par exemple, quand le poids algorithmique de certaines comparaisons entre images est trop important, il est logique de pré-filtrer par des descripteurs simples un petit ensemble d'images dans lequel les comparaisons coûteuses pourront avoir lieu. Mais le pré-filtrage peut avoir de nombreux autres sens. Simplifier un domaine pour en extraire plus facilement les similarités peut prendre d'autres formes. On peut penser ici à un pré-filtrage textuel basé sur des mots clés. Mais aussi, si l'on est capable d'extraire des méta données bas niveaux sur une image (intérieur/extérieur, visage/non visage, plan rapproché/plan éloigné…) alors, on peut se servir de celles-ci en tant que pré-filtrage. Éliminer des images qui de toute façon ne sont pas solutions, ne peut être que bénéfique pour la véritable phase de recherche avale.

1.7.4 Contrôle de pertinence

Sans entrer véritablement dans les détails des méthodes de rappel de pertinence (relevance feedback en anglais), le jugement de l'utilisateur post-recherche est très utilisé dans divers domaines d'apprentissage automatique, comme la recherche textuelle par exemple. En effet, l'utilisateur peut donner une information de type Satisfait / Insatisfait ou alors évaluer son contentement via une échelle graduée suivant la correspondance entre l'image "résultat" et ce qu'il attendait réellement. Le moteur de recherche, connaissant l'appréciation de l'utilisateur aux réponses fournies, peut ainsi améliorer la recherche. Diverses possibilités, non exclusives, s'offrent au système [Blancho et Konik, 2000, Sciascio *et al.*, 1999] :

- Affiner la recherche en cours : les images positives / négatives permettent de pondérer le poids des différents descripteurs dans la recherche.
- Créer un profil pour chaque utilisateur: l'ajout des différentes requêtes et leur appréciation permet de générer un profil de l'utilisateur, utilisé ensuite pour chaque requête. Le profil est généralement constitué de l'ensemble des poids affectés à chaque descripteur.
- Générer une pondération améliorant la recherche: similaire au profil utilisateur mais global à l'ensemble des utilisateurs.

Le jeu de pondération, résultant d'une technique de "relevance feedback", est fortement lié au domaine de la base d'images. En effet, une pondération, bien qu'efficace sur un ensemble d'images, n'est plus forcément adaptée sur une autre base d'images. Diverses méthodes de calcul de la nouvelle pondération existent, soit à partir des éléments pertinents, soit à partir de ceux non pertinents.

Néanmoins, les techniques de "relevance feedback", si elles ne sont pas à exclure, sont difficiles à appréhender pour l'indexation d'images. En effet, il est clair que cela ne peut qu'apporter un plus au moteur de recherche mais un mauvais descripteur restera toujours un mauvais descripteur, contrôle de pertinence ou non. Plus précisément, il serait présomptueux de penser qu'un rappel de pertinence, basé sur les descripteurs actuels, puisse suffire à répondre aux attentes des utilisateurs de bases généralistes. La pertinence des réponses ne permettra d'améliorer le système que si celui si est déjà performant du point de vue de l'utilisateur, d'où notamment leur utilisation dans des domaines bien précis [Ciocca et Schettini, 2004].

1.7.5 Divers systèmes existants

De nombreux systèmes ont été développés pour la recherche d'images par le contenu. Qu'ils soient utilisés par les professionnels ou dans le domaine universitaire, ces systèmes ont été la source de nombreux articles et études. Parcourons brièvement les systèmes CBIR [7] les plus marquants depuis les années 90.

1.7.5.1 QBIC

QBIC [Flickner *et al.*, 1995], Query By Image Content, est le premier système commercial de recherche d'images par le contenu, il est aussi de loin le plus connu. Il a été développé au sein de la société IBM, et a été utilisé par de nombreuses sociétés. Une des volontés des auteurs de QBIC fut de rassembler un large panel de descripteurs d'informations contour, couleur et texture. Ainsi le système met en oeuvre la majorité des différentes approches en indexation d'images.

En omettant le rappel de pertinence, nous pouvons dire que QBIC implémante les principales caractéristiques d'un système CBIR :

- Descripteurs couleur, texture, forme. Ces descripteurs sont des descripteurs classiques, nous en avons précédemment cité certains. Les approches forme et texture sont assurées notamment par des signatures visuelles classiques (contraste, directionnalité, entre autres pour la texture; circularité, surface, ... pour la forme).
- Différents types de requêtes : par l'exemple, par croquis ou par mixité exemple/croquis - les contours des images sont extraits via le détecteur de contour de Canny [Canny, 1986].
- Une véritable indexation multidimensionnelle des paramètres grâce aux $R^* - tree$, un modèle d'arbre de recherche très utilisé dans les bases de données.
- Les résultats sont classiquement présentés du plus proche au plus éloigné.

QBIC fait maintenant partie de l'extension multimédia (DB2 AIV Extenders) du système de gestion de bases de données DB2, développé aussi par IBM. Cette extension inclut non seulement

7. Content Based Image Retrieval

des outils de recherche d'images (via SQL donc) mais aussi de vidéos et d'audio. Malheureusement, les apports récents apportés au système de recherche ne font pas l'objet de publications scientifiques de la part d'IBM.

1.7.5.2 VisualSeek

VisualSeek [Smith et Chang, 1996] est un des premiers CBIR basé sur une approche région, via une quantification couleur de ces régions dont les couleurs dominantes sont extraites. Le pavage de chaque image par les couleurs dominantes permet d'établir des relations spatiales : la recherche consiste à retrouver des régions de même couleur dominante avec la même distribution spatiale. Plus encore, le système propose différents types de requêtes: par région unique, par région multiples...

La quantification utilisée dans VisualSeek est réalisée dans l'espace HSV. L'espace couleur est ainsi divisé en 18 teintes, 3 saturations, 3 Luminance et 4 gris, soit 166 classes ($18 \times 3 \times 3 + 4$)

1.7.5.3 GNU Image-Finding Tool

Le système GIFT [Müller, 2001], GNU Image-Finding Tool, et le protocole pour les communications clients-serveur MRML [Müller *et al.*, 2003] sont en fait l'extension du sytème Viper, developpé au sein de l'université de Genève. Le but de ce système est de proposer une architecture moderne, sur laquelle il est possible de greffer des plugins propres. Néanmoins, cette tentative de CBIR moderne et évolutif n'a pas connu le succès escompté et le projet est arrêté depuis 2003.

1.7.5.4 Netra

Netra [Ma et Manjunath, 1999a] développé au sein de l'université de Santa Barbara, se propose de naviguer dans de larges bases d'images. Pour cela une segmentation permet de calculer des attributs couleur, forme et texture. C'est un système assez classique tant du point de vue de l'indexation (SS-tree) que du calcul des distances (euclidiennes). Néanmoins il s'agit sans doute du sytème purement universitaire le plus abouti.

1.7.5.5 SIMPLIcity

Le système simplicity [Wang *et al.*, 2001], Semantics-Sensitive Integrated Matching for Picture LIbraries est basé sur la volonté de mixer des informations sémantiques de type intérieur / extérieur avec des informations statistiques issues d'une segmentation. Ainsi les images sont automatiquement triées suivant des critères sémantiques simples, ce qui permet ensuite d'accélérer et d'aider la recherche d'images similaires. Ce sytème a été abandonné en 2001 et n'est

plus développé actuellement. Néanmoins une démonstration en ligne parfois fort convaincante est encore disponible à "http://wang.ist.psu.edu/IMAGE/" .

1.7.5.6 Virage

Virage [Bach *et al.*, 1996] n'est pas à proprement dit un système complet de recherche par le contenu. L'objectif initial était de construire un environnement dédié à la recherche d'images, principalement composé de primitives. Une primitive est composée d'un attribut et d'un calcul de distance sur cet attribut. Le sytème propose aussi différents formats de données et la possibilité d'ajouter de nouvelles primitives. L'interface classique donne la possiblité de pondérer les différentes primitives ainsi que d'utiliser le bouclage de pertinence. Virage, de la société Virage Inc, s'il n'était pas novateur par rapport à QBIC en 1996, avait l'avantage d'une ouverture et de n'être pas qu'un système, mais aussi un ensemble réutilisable de descripteurs. Notons que Altavista Photofinder a opté pour Virage comme moteur de recherche interne.

1.7.5.7 Blobworld

Le système Blobworld [Carson *et al.*, 1999] se donne pour but de retrouver, à partir d'une image requête, des régions similaires en couleur et texture, appelés blobs, dans les images de la base. Ainsi le choix d'un nombre limité de blobs dans l'image source permet de retrouver ces blobs dans la base d'images. Bien sûr, c'est à l'utilisateur de choisir les régions les plus représentatives de son image requête. Pour cela, il choisit par exemple 2 ou 3 régions qu'il juge importantes dans une image segmentée. C'est une approche intéressante dans le pari, fait par les auteurs, que retrouver une partie de l'information couleur et texture permet de retrouver des images similaires.

1.7.5.8 Et encore de nombreux autres...

On trouve dans la littérature de nombreuses études. Citons notamment l'étude [8] ,faite en 2001, par R. Veltkamp and M. Tanase[Veltkamp et Tanase, 2002] se proposant, d'énumérer les caractéristiques des principaux systèmes connus. Néanmoinsn peu sont encore actifs actuellement, la plupart n'étant plus développés. Tout simplement pris dans le piège de l'exhaustivité et de la complexité des requêtes et des attentes de l'utilisateur final, aucun ne permettait de satisfaire totalement ce dernier.

8. "http://www.aa-lab.cs.uu.nl/cbirsurvey/cbir-survey/cbir-survey.html"

1.7.6 Le sytème *i*COBRA

Durant la première étape de cette thèse, afin de mieux appréhender la problèmatique même de la recherche d'images par le contenu, il nous a semblé fort constructif d'implémenter un moteur dont nous maîtriserions les tenants et aboutissants. Toutefois, notre ambition n'était pas de combler les lacunes des systèmes CBIR existants mais avant tout de pouvoir opposer ou critiquer certains descripteurs à la réalité. Ainsi, j'ai développé ***i***COBRA, pour Image COntent Based Retrieval Application, au sein du laboratoire. Voyons maintenant brièvement les bases de ce système que l'on peut trouver à l'adresse http://www.ligiv.org/icobra/ .

- Interface en ligne. Basée sur une combinaison de Javascript, PHP et HTML, l'interface est ainsi accessible en ligne depuis n'importe quel navigateur internet.
- Plusieurs centaines de milliers d'images sont intégrées la base.
- Une SGBD permet de stocker les informations relatives aux images et les descripteurs calculés.
- Une interface pour l'indexation d'images où différentes actions sont proposées.
- Un partie privée permettant de réaliser des mesures de classifications d'images. À partir de classes pré établies, cette partie permet de calculer pour un descripteur particulier les différentes statistiques relatives à la classification: Taux de reconnaissance, précision, rappel…
- Une totale et aisée automatisation de l'ajout et de la suppression d'un descripteur.
- Une interface pour la visualisation couleur et l'application d'algorithmes.
- Une interface pour les espaces hybrides couleur décorrélés [9].

On trouvera en annexe A une description plus détaillée des différentes parties d'***i***COBRA.

9. Ce travail a fait l'objet d'une publication : [Da Rugna *et al.*, 2004]

Chapitre 2

UN CONSTAT D'ÉCHEC ?

Sommaire

Les systèmes de recherche d'images par le contenu, aussi complexes soient-ils, ne coïncident pas avec les attentes concrètes des utilisateurs. Ce constat oblige assurément à recentrer la problématique de l'indexation d'images et à remettre en questions les techniques utilisées. Dans ce contexte, nous exposons alors les différentes voies que nous développons dans ce manuscrit.

Loin de l'engouement de la fin des années 90, ces dernières années ont été le cadre d'un important repli du domaine de la recherche d'images par le contenu. Faut-il y voir un abandon face à des difficultés trop grandes ou un simple effet de mode passé ? Finalement, la gestion des informations visuelles ne souffre t'elle pas des mêmes maux qu'ont pour souffrir d'autres domaines du traitement d'images comme l'analyse Jolion en 1998[Jolion, 1998]? Pour étayer notre point de vue, nous allons axer notre discours sur deux points. Après avoir resitué les avancées du domaine par rapport aux mots clés notamment, puis recentré le débat sur les attentes de l'utilisateur, on conclura sur les ouvertures vers l'aspect sémantique.

2.1 De l'usage des mots clés à l'extraction automatique

Indexer une base d'images par mots clés est certes très coûteux en temps humain mais, et ce n'est pas négligeable, cela satisfait généralement pleinement l'utilisateur. Même dans les cas extrêmes où des millions d'images sont référencées par leur nom de fichier, il est possible de trouver l'image recherchée. Qui n'a pas utilisé "Google images" [1] et trouvé rapidement dans de nombreux cas l'image voulue... En pratique, le piège de l'exhaustivité existe mais en définitive les utilisateurs savent composer avec ce problème. Dans le cadre d'images généralistes, comme une base d'agences de presse, les mots clés sont souvent très détaillés comme le montre la figure2.1 [2] . Même s'ils sont largement insuffisants pour combler toutes les attentes de l'utilisateur, il est indéniable que l'information contenue dans ces mots clés est très fournie. De plus, certains de ces mots comme "Madrid" ou "Spain" sont vecteurs du contexte de la prise d'image, contexte qu'il est impossible d'extraire de l'image seule.

Adults
Agriculture
animal
breeding
combat uniform
demonstration for
demonstrator
demonstrator attitude
dog
Europe
Europeans
herd
holding
hook
in the street
job in Spain
Madrid
Males
man face
man job
man position
Men
Occupations and work
ovine
people of Spain
Photography
Political and social issues
Protest
Photography
Political and social issues
Protest
sheep
shepherd
social action
social issues in Spain
society and tradition
Spain
Spaniards
Spanish child
standing
stick
street
traditional profession
transhumance
unusual
photo Women

© DESPOTOVIC DUSKO/CORBIS SYGMA

Fig. 2.1 – *Exemples de mots clés de la base photographique Corbis*

1. http://www.google.fr/imghp
2. Corbis ©http://www.corbis.com

Que peuvent alors apporter dans ce cadre les méthodes d’extraction automatique de connaissance à partir du seul contenu pixelique des images ? Il est néanmoins notable que ces 20 dernières années ont vu un essor considérable du domaine. L’efficacité des méthodes de recherche a de ce fait aussi été considérablement améliorée. Rechercher un objet sur fond blanc dans une base d’objets sur fond blanc n’est plus de nos jours une problématique. La large batterie de descripteurs que nous maîtrisons permet d’apporter une solution satisfaisante. De même, il est possible de retrouver efficacement des images dans des cas bien particuliers, comme retrouver par exemple des “couchers de soleil” dans une base généraliste.

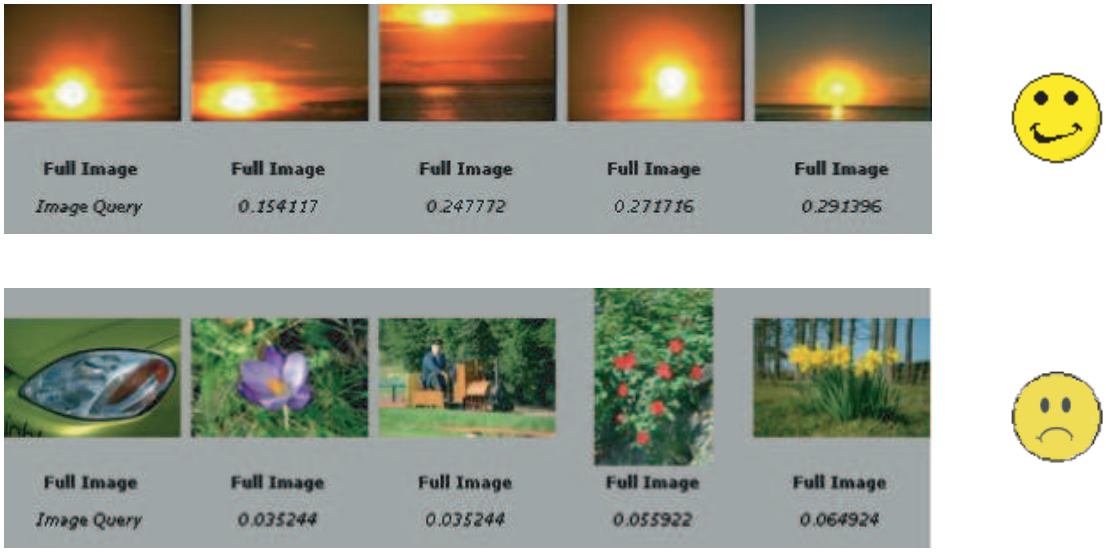


Fig. 2.2 – *Pas sémantique infranchissable ?*

Pourtant, tout ce panel de descripteurs et de systèmes de recherche par le contenu n’a pas remplacé les sytèmes par mots clés. La grande majorité des personnes travaillant avec des photographies ou des vidéos est en effet sous la contrainte de devoir annoter les images. Pour une raison simple: la recherche par mots clés fonctionne et celle par le contenu ne fonctionne pas. Il peut sembler radical de s’exprimer ainsi mais l’utilisateur lambda ne connait pas l’expression “preque bon” quand il exécute une recherche d’images. Le pas sémantique entre l’utilisateur et le moteur de recherche comme illustré sur la figure 2.2 est un fossé beaucoup trop large. Retrouver des notions simples est possible mais reconnaître un phare de voiture dans l’image du dessous est fort difficile dans un contexte généraliste, c’est-à-dire sans préciser que la base d’images est une base de voitures par exemple. Appréhender la sémantique d’une image n’est pas encore possible à partir du simple contenu…

Doit-on donc tirer un trait sur la recherche d’images par le contenu ? Certainement pas! Sans doute, et cela est déjà le cas, il faut certes repositionner la manière d’utiliser les descripteurs et les méthodes de recherche. Il faut notamment aller là ou les mots clés ne peuvent de toute évidence

pas aller. Chaque image d'une base, même indexée par 100 mots, ne pourra pas représenter précisément toutes les parties de l'image et toutes ses caractéristiques. Sur une image où les mots clés définissent un journaliste en train d'interviewer un homme politique, l'information "couleur de la chemise de l'homme politique" par exemple n'est pas enregistrée. Ainsi, les outils que la communauté a élaborés au cours de ces années peuvent naturellement s'intégrer et ainsi apporter un plus relatif à la description par multiples mots-clés. Encore faudrait-il introduire des modèles fédérateurs de gestion de connaissances[Santini, 2001]…

2.2 La nécessaire étape de segmentation

Nous l'avons vu, segmenter une image en zones afin d'en étudier celles-ci est une étape obligatoire dans la compréhension d'une scène. Au demeurant, on reproche dans de nombreux cas à l'indexation d'images des travers qui viennent objectivement de cette étape. Les résultats de cette dernière sont en effet encore loin des espoirs mis en eux par le système.

La notion même de contenu d'une image se doit alors d'être abordée[Santini, 2001]. La figure 2.3 en est un illustration: non seulement une image n'est pas toujours segmentable mais qui plus est, son interprétation a-t'elle un sens ? Il est paradoxal d'attendre un haut niveau sémantique d'une opération finalement bas niveau.

Fig. 2.3 – *Paradoxe de la segmentation ?*

Malgré tout que doit-on alors véritablement attendre de la segmentation ? La stabilité au contexte ? Un découpage en zones sémantiques ou en entités insécables ? Un découpage en zones homogènes ? Segmenter une image est sans doute l'étape la plus importante de l'extraction des connaissances. Si la segmentation ne fournit pas ce que l'on attend, alors le descripteur ne pourra pas être véritablement discriminant. Dans une optique indexation d'images, les connaissances objectives sur les méthodes de segmentation sont rares. Par le fait, certaines approches utilisent des relations spatiales entre régions, d'autres codent les formes. Mais, la remise en question du résultat même de la segmentation n'est pas faite. Par exemple, retrouver un objet dans plusieurs

scènes conjecture que cet objet est segmenté quasi-similairement. Aussi, notre travail présenté en partie II s’est axé sur la mise en place d’une évaluation objective de ces méthodes de segmentation dans un contexte orienté indexation d’images.

2.3 Où l’on reparle de sémantique

Si l’on considère que la recherche d’images par le contenu va permettre d’affiner une recherche préalablement effectuée par mots clés notamment, cela induit le fait qu’une recherche d’images “orientées” répond à l’attente sémantique de l’utilisateur. Cela signifie que si, pour une image, on possède un ensemble d’informations, il sera alors possible de réaliser une recherche plus performante. Avant même de réaliser la recherche par le contenu proprement dite, il est donc nécessaire d’accumuler de l’information sur la requête. On parle souvent de “sémantique induite” pour définir cette information retrouvée indirectement par une méthode. De notre point de vue, une étape d’extraction de pré-connaissance, des méta-données, est nécessaire avant même d’essayer de réaliser une indexation. Il faut capitaliser le plus grand nombre d’informations primaires sur une image: intérieur/extérieur, avec mouvement/sans mouvement, de jour/de nuit. . . De cette information recueillie par des traitements bas niveaux, il devient possible d’extrapoler sur l’ensemble des descripteurs ceux qui vont être pertinents et ainsi permettre une réponse adaptée.

Fig. 2.4 – *Comment voit-on cette image?*

Prenons l’exemple ici de l’image 2.4. Regarder, et donc analyser cette image, nous permet de reconnaître un petit animal posé sur un rocher, sans doute devant une forêt qui constitue le fond de l’image. Dans cette image, nous sommes sûrs que le photographe a voulu montrer principalement l’écureuil. Sans entrer dans la perception même de cette image, un facteur important nous a aidé à nous situer spatialement et à reconnaître la zone importante de l’image: le flou et la profondeur qui en résulte. Cette exemple montre ainsi que l’information floue ne peut qu’apporter un plus fortement sémantique dans une approche indexation. Caractériser différentes zones d’une image

et leur donner différents poids sémantiques (flous/non flous par exemple) sera une méta-donnée capable d'aider à la recherche d'images. Avant même d'extrapoler une utilisation directe, il nous a alors semblé judicieux de proposer une extraction de zones floues et de confronter celle-ci face à une base généraliste.

Par ailleurs, si l'étape de segmentation ne permet pas d'extraire des objets au sens sémantique du terme, comment se servir des ces régions extraites. Blobworld[Carson *et al.*, 1999] proposait par exemple d'établir une similarité entre images en retrouvant un ensemble de régions de l'image initiale. En extrapolant cette vision zones d'intérêt, peut-on extraire des régions émergentes d'une image ? Si oui, retrouver ces régions émergentes a-t'il un sens ? Bien-sûr, il ne s'agit pas de créer une véritable mesure de similarité entre images mais de proposer une extraction purement bas niveau, donc non sémantique, puis de montrer que son utilisation peut apporter un critère de similarité maîtrisable.

Parallèlement, d'un point de vue théorique, on peut se poser une question basique: connaissant la requête et connaissant les différents descripteurs calculés sur la base d'images, comment répondre au mieux à la requête ? En effet, un ensemble de descripteurs disparates et nombreux rend réellement l'association de ceux-ci difficile afin d'apporter une bonne réponse. Comme le notaient les auteurs dans [Santini et Jain, 1997], "Images databases are not databases with images". Il convient sans doute que le modèle où l'on désire exécuter les requêtes puisse prendre en compte les caractéristiques propres du descripteur proposé. Recomposer, par exemple, un, ou plusieurs descripteurs, afin d'en créer un nouveau nécessite un modèle et une vision moins restreinte que celle fournie pas les bases de données classiques. Dans ce contexte, nous proposons par exemple une algèbre d'histogrammes qui permettra alors de tenir compte des spécificités de ceux-ci.

Deuxième partie

Segmentation d’images couleur

Chapitre 1

MÉTHODES DE SEGMENTATION: ÉTAT DE L'ART

Sommaire

Segmenter une image est certainement un des sujets les plus étudiés, comme en témoigne le nombre de publications relatives disponibles. Ce chapitre se propose alors de décrire brièvement les méthodes de segmentations classiquement utilisées dans les moteurs de recherches. Ensuite, nous proposerons une approche par suivi pyramidal dont la segmentation grossière résultante servira de base à d'autres études décrites plus en aval.

1.1 Méthodes usuelles

Sans nous préoccuper pour l'instant de la problématique même de la définition d'une segmentation, admettons néanmoins qu'il est délicat d'espérer définir la bonne segmentation pour une image de scène. On peut attendre en effet d'une part des zones homogènes en couleur, d'autre part des zones homogènes en texture ou bien encore une segmentation définie par le mouvement de la scène. C'est dans ce contexte que nous nous sommes alors intéressés à différentes méthodes de segmentation. Grossièrement, ces dernières utilisent les informations suivantes:

- le nuage couleur, soit directement, soit indirectement comme les histogrammes basés sur une projection/discrétisation;
- un traitement basé sur une approche morphologique. Il s'agit d'analyser l'image en tant que surface par exemple pour extraire les différentes régions qui la composent, en trouvant des contours ou des bassins importants. On parlera ici d'approche région ou d'approche contour.

Certaines méthodes mixent les deux approches soit conjointement, soit successivement afin d'exclure les problèmes d'artéfacts dus à du bruit, par exemple. Néanmoins une caractéristique générale ressort dans les méthodes actuelles : la meilleure utilisation possible de l'information couleur [Trémeau *et al.*, 2004]. Longtemps négligée, cette information n'avait pas encore été exploitée correctement jusqu'à ces dernières années, de nombreuses méthodes "couleur" se réduisant souvent à l'approche marginale. L'algorithme est étendu aux trois canaux séparément et les résultats sont ensuite fusionnés. La couleur est maintenant une véritable information et parler de segmentation couleur n'est plus d'actualité dans le monde de l'indexation d'images, tant la couleur est devenue un élément naturel des méthodes classiques dans le domaine.

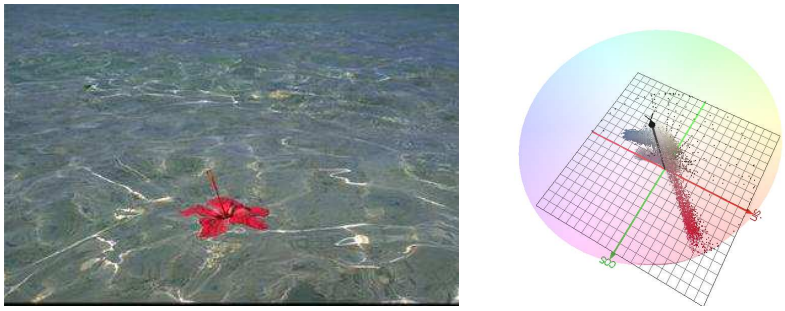
Cette phase de segmentation étant éminemment sensible, elle fait l'objet de nombreux articles dans la littérature. Plutôt que de développer les diverses techniques mises en jeu, citons les différentes synthèses des études traitant de la couleur depuis [Pal et Pal, 1993] jusqu'aux références [Cheng *et al.*, 2001] et [Lucchese et Mitra, 2001]. En principe et de façon grossière, il n'est pas inhabituel de regrouper l'ensemble des méthodes existantes en trois familles principales :

- des méthodes de partitionnement de l'espace de paramètres (clustering, k-mean, histogramme);
- des méthodes orientées espace image (split-and-merge, croissance de région, contours, réseaux neuronaux);
- des méthodes d'ordre physique ([Maxwell et Shafer, 2000]).

bien sûr, certaines méthodes [Rezaee *et al.*, 2000] reposent sur la combinaison de plusieurs de ces approches ou mettent à profit des paramètres couleur combinés avec des descripteurs d'ordre texturel [Deng et Manjunath, 2001, Chen *et al.*, 2002].

Après avoir présenté, de façon non exhaustive, différentes approches de segmentation d'images, nous verrons les méthodes classiquement utilisées dans les moteurs de recherche. Finalement, nous présenterons une nouvelle adaptation des approches pyramidales dont l'objectif est d'obtenir des segmentations grossières d'images de scènes.

1.1.1 Segmentation par analyse du nuage colorimétrique



(a) Image originale (b) Représentation HSV

Fig. 1.1 – *Nuage couleur clairement séparable*

C'est sans doute dans cette approche que l'on dispose du plus large panel, dont de nombreuses optimisations ou adaptations. D'abord un constat simple : si l'on regarde le nuage couleur d'une image, illustré figure 1.1, il se dégage souvent l'impression que la dispersion colorimétrique permet de retrouver les objets initiaux. Sur cet exemple, on voit clairement que dans l'espace HSV, il est possible de distinguer parfaitement deux nuages couleurs : les rouges et les autres. Or cela correspond à ce que l'on attendrait d'une segmentation couleur : la fleur et l'eau. Néanmoins cet exemple simpliste nous permet d'exhiber les points critiques que nous rencontrons dans ce type d'approche :

- L'espace couleur.
  Le choix de l'espace couleur va influencer directement la qualité de la segmentation. Tel algorithme fonctionnera mieux sur tel type d'images en choisissant $L^*a^*b^*$ par exemple. Un autre sera plus adapté avec un espace comme $I_1I_2I_3$. Diverses études ont été menées ces dernières années sur les avantages et inconvénients des différents espaces colorimétriques [Lew, 2001], [Lee *et al.*, 1994] mais aucun espace ne semble sortir du lot dans un contexte généraliste.
- Le codage de l'information.
  Si on imagine une représentation classique d'une image en un triplet couleur type RGB, on doit s'interroger sur la quantité d'information proposée. En utilisant des projections

ou des discrétisations plus fortes, on compresse l'information sous une autre forme plus discriminante. C'est pour cela que les histogrammes sont si importants : en présentant différemment l'information, avec perte néanmoins, des algorithmes simples peuvent aboutir à une segmentation correcte de l'image.

- Les artefacts locaux.
  Une fois les couleurs étiquetées, chaque pixel est étiqueté comme étant de même label que sa couleur. Ceci permet de reconstituer les régions: c'est l'ensemble des pixels adjacents de même étiquette. Cela peut engendrer bien sûr des artéfacts qui sont dus notamment aux dégradés, frontières entre régions, petits objets, bruits. Souvent une étape de suppression des petites régions résout localement ce problème. On parle alors d'algorithme post-segmentation.

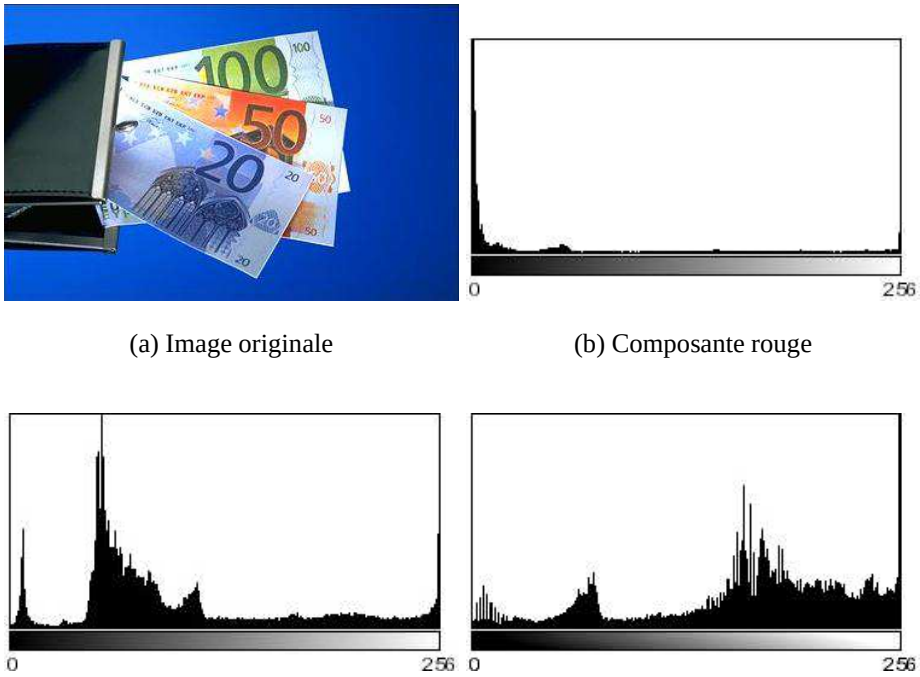
1.1.1.1 Segmentation par histogrammes

Il est possible de modéliser les histogrammes couleurs comme un ensemble de 3 histogrammes à une dimension ou bien comme un histogramme ou chaque couleur est sur 3 dimensions. On pourrait aussi exposer de nombreuses variantes intéressantes, comme notamment les histogrammes spatiaux chromatiques flous [Lambert et Grecu, 2003].

1.1.1.1.1 Histogrammes 1D Le seuillage d'une composante couleur, souvent par recherche de pics, permet de séparer les pixels en sous-ensembles. Comme le montre la figure 1.2, chaque histogramme 1D possède ses propres seuils et permet une classification en différentes classes. Notons différents types d'approches pour réaliser une segmentation :

- Mono dimensionnel - On ne prend en compte qu'un seul histogramme 1D, celui où l'information est la plus importante.
- Récursif - On segmente par technique multi-seuillage [Lambert et Macaire, 2000] en utilisant itérativement l'histogramme permettant a priori la séparation la plus efficace en deux classes [Ohta *et al.*, 1980].
- Hiérarchique - On utilise un multi-seuillage [Cheng, 2000] mais à l'intérieur de chaque région trouvée. Les régions extraites du premier seuillage (souvent sur la teinte) sont ensuite segmentées séparément par histogramme 1D. Ainsi des régions fortement homogènes dès le premier seuillage ne seront plus éclatées. Cela permet ainsi d'ajouter un côté local au seuillage par histogramme. La figure 1.3 montre deux segmentations issues de l'algorithme proposé dans [Cheng, 2000].

Bien évidemment, comme décrit dans de nombreuses études [J.P. Cocquerez, 1995], cette rapide description laisse entrevoir différents points litigieux : premièrement le type de seuillage à appliquer et le jugement de son efficacité. En effet, un critère d'arrêt doit être mis en place



(a) Image originale (b) Composante rouge

(c) Composante verte (d) Composante bleue

Fig. 1.2 – *Histogramme RGB.*

Fig. 1.3 – *Exemples de segmentations par histogramme hiérarchique*

pour empêcher une sur-segmentation. De plus, il est aussi nécessaire de juger l'efficacité d'un seuillage pour choisir la composante sur laquelle il sera appliqué. Il s'agit alors de paramètres statistiques très affinés. Deuxièmement, il convient de remarquer que l'espace RGB n'est sans doute pas le plus adapté pour l'application des méthodes d'histogrammes 1D. D'autres espaces, comme l'espace $X_1X_2X_3$, HSV ou même encore $L^*u^*v^*$[Schettini, 1993] sont parfois utilisés et fournissent des résultats probants.

1.1.1.1.2 Histogrammes 3D Il existe peu d'algorithmes utilisant directement une vision en 3 dimensions des histogrammes, ceci étant dû à la taille très importante d'un nuage couleur 3D. Pourtant un nuage couleur peut être séparable en 3 dimensions sans l'être de manière évidente sur un seul axe. Les méthodes les plus classiques utilisent une détection de pics dans l'histogramme 3D, souvent par une approche morphologique [Park *et al.*, 1998]. Ceci permet ensuite de séparer les pixels entre ces pics.

1.1.1.2 Clustering - Nuées dynamiques

Il existe, dans la littérature des modèles statistiques [Duda *et al.*, 2001], un nombre très important de méthodes de clustering d'un nuage de points en 3 dimensions. On peut penser aux méthodes coopératives, aux nuées dynamiques, aux nuées dynamiques floues ou encore aux méthodes par graphes d'adjacence [Tremeau et Colantoni, 2000]. Nous présentons ici la méthode la plus générique et la plus utilisée : l'approche par nuées dynamiques. Cela consiste tout d'abord dans le choix d'un nombre fixé de germes. Ensuite, chaque pixel est affecté au germe qui lui est le plus proche, ce qui permet de constituer une classe. Chaque classe permet de recalculer un nouveau germe. Ce processus est itéré jusqu'à convergence. Cet algorithme est adaptable à de nombreux cas de figures. En fait, on peut appliquer une méthode de classification par nuées dynamiques à n'importe quel ensemble de pixels muni d'une distance entre eux. En outre, c'est un algorithme relativement facile à mettre en oeuvre mais son principal défaut est sans doute sa faible capacité à gérer directement des ensembles très importants et très disparates de pixels (plusieurs millions de couleurs par exemple).

1.1.1.3 Mean Shift

À l'origine, le mean shift a été introduit afin de réaliser le lissage d'images par moyennage. La procédure mean shift [Cheng, 1995] est une procédure itérative de recherche de maxima locaux dans un espace, basée sur une montée de gradient. Une utilisation efficace de cette procédure, pour extraire des clusters, dans le nuage couleur avec un objectif de quantification et de segmentation est proposée dans [Comaniciu et Meer, 2001]. Les résultats sont généralement

Algorithme 2: Classification par nuées dynamiques

Données : Ensemble de couleurs

Choisir k germes distincts de 1 à k, noté germe[i];
répéter
    evolution = faux;
    pour chaque *couleur* n **faire**
        calcul du germe i le plus proche de n;
        classe[n] = germe[i];
    fin
    pour chaque *classe* i **faire**
        centre[i] = centre des couleurs de classe i;
    fin
    pour chaque *classe* i **faire**
        si *centre[i]* ! $=$ *germe[i]* **alors**
            evolution = vrai;
            centre[i] = germe[i];
        fin
    fin
jusqu'à *evolution* ! $=$ *vrai*;
Résultat : Classification de chaque couleur

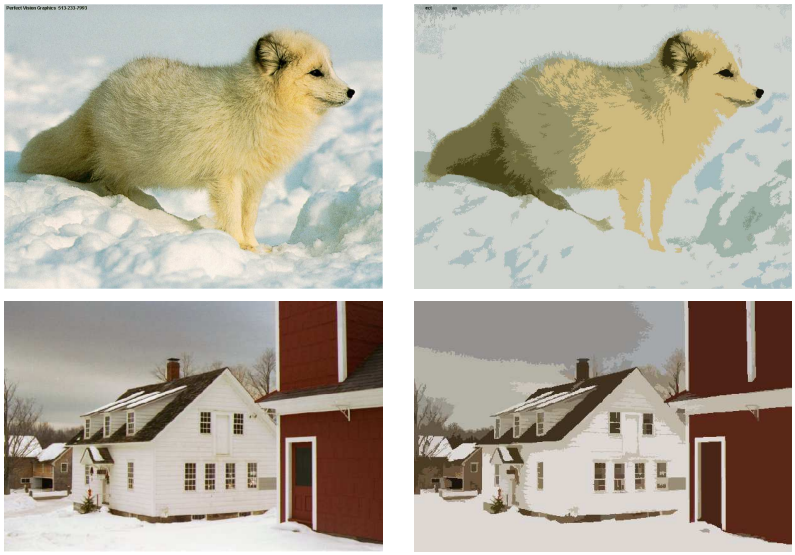


Fig. 1.4 – *Exemples de segmentations par nuées dynamiques (20 classes)*

graphiquement très convaincants. Néanmoins, un grand nombre de régions est en général extrait par cette méthode.

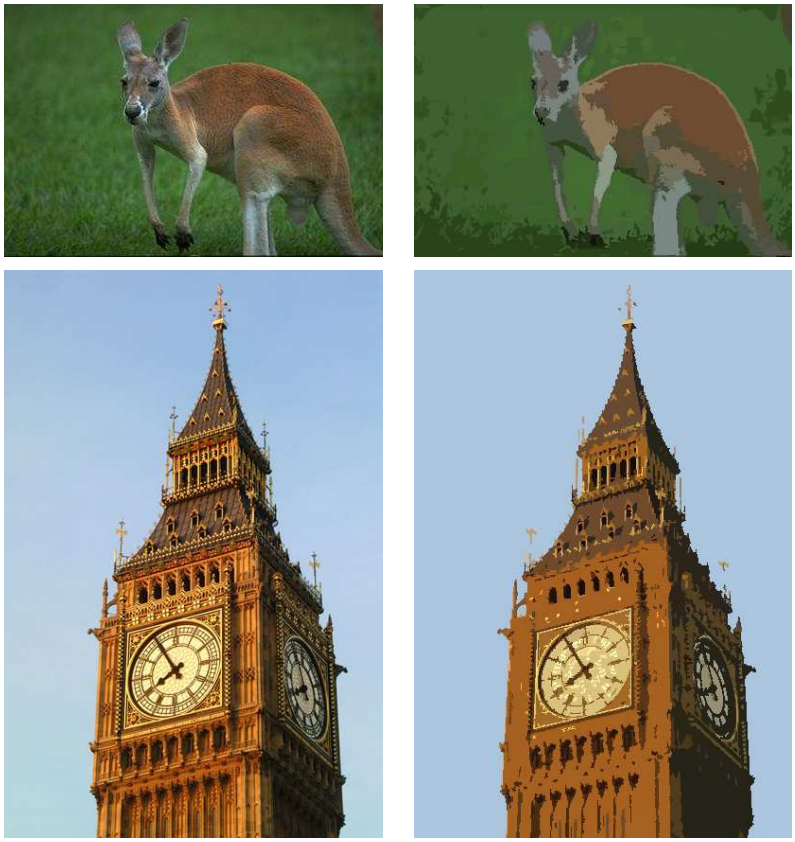


Fig. 1.5 – *Exemples de segmentations par l'algorithme Mean shift.*

1.1.2 Segmentation par approche région

À l'opposé des méthodes "pixeliques", les méthodes dites par régions sont basées sur l'aspect homogène d'une région. L'homogénéité peut être définie par une même couleur, une même texture, un même effet (de flou par exemple) ou encore un même objet. Il s'agit d'utiliser l'information spatiale conjointement à l'information colorimétrique pour obtenir la classification en régions de l'image [J.P. Cocquerez, 1995]. Bien sûr, les plus anciennes et les plus classiques approches sont celles de type fusion ou division. Pour segmenter l'image en régions homogènes, on utilise soit un algorithme de fusion qui relie les régions entres elles, soit un algorithme de division qui éclate les régions jusqu'à l'obtention d'une certaine stabilité. Notons aussi la possiblité de mixer les deux approches afin d'atteindre une stabilité par miminum ou optimum.

Une autre méthode classique est la croissance de régions [Chassery et Garbay, 1984] à partir d'un ensemble de germes auxquels sont agrégés les pixels adjacents, jusqu'à un critère d'arrêt.

La croissance de régions est sans doute une des méthodes la plus utilisée et qui possède un très grand nombre de variantes, que cela soit sur le type d'agrégation ou bien sûr, sur le critère d'arrêt. D'autres approches sont aussi notablement adaptées aux images de scènes: l'approche partage des eaux et l'approche pyramidale.

1.1.2.1 Ligne de partage des eaux

En partant d'une analogie avec un relief terrestre, un watershed est en fait une région où "s'écoulerait" l'eau. L'apparition d'un lac au cours de l'inondation est due à la présence d'un minimum local dans le gradient de l'image. C'est pourquoi, lors de l'élaboration de l'algorithme, on place un attracteur au niveau de chaque minimum local. Un attracteur absorbe alors progressivement tous les pixels, se trouvant dans la vallée qui l'entoure, dans le relief accidenté. Lorsque la croissance des attracteurs est terminée, l'image est segmentée en régions homogènes [Vincent et Soille, 1991, Kim et Kim, 2003] . Le problème majeur rencontré est la sur-segmentation engendrée. Pour la réduire, plusieurs techniques sont possibles. On peut appliquer des filtres réducteurs de bruit ou utiliser la notion de profondeur des bassins dans l'algorithme. Il est alors possible d'analyser les régions obtenues afin de réduire la sur-segmentation. La figure 1.6 illustre une segmentation par WaterShed comme défini dans [de Andrade *et al.*, 1999]. Des algorithmes de segmentation basés sur un WaterShed utilisant des approches multi-échelles [Dombre, 2003, Vanhamel *et al.*, 2001] ou floues [Philipp-Foliguet et Lekkat, 2004] ont été mis en place spécifiquement dans le cadre de la recherche d'images. Ils présentent des qualités propres, qui les rendent plus efficients sur certaines bases d'images.

Fig. 1.6 – *Exemples de segmentations par segmentation watershed.*

1.1.3 Approche pyramidale

Les techniques multirésolutions [Deng et Manjunath, 2001, Bimbo *et al.*, 1998], quand bien même les critiques relatives à l'outil lui-même sont référencées ([Bister *et al.*, 1990]) sont souvent utilisées pour leur rapidité et leur polyvalence. À l'origine, Burt et Rosenfeld ont proposé en 1981 une technique multirésolution de segmentation d'images [Burt *et al.*, 1981]. Depuis, cette méthode de suivi de germes le long de la pyramide a engendré de nombreuses adaptations [Konik, 1994, Ziliani et Jensen, 1998, Fuh *et al.*, 2000, Rezaee *et al.*, 2000, Grau *et al.*, 2004]. La figure 1.7, issu de [Marfil *et al.*, 2004] montre des résultats de la méthode proposée par les auteurs (BIP) face à diverses méthodes: pyramide adaptative [Jolion et Montanvert, 1992], pyramide pondérée [Prewer et Kitchen, 2001] et pyramide liée [Bister *et al.*, 1990].

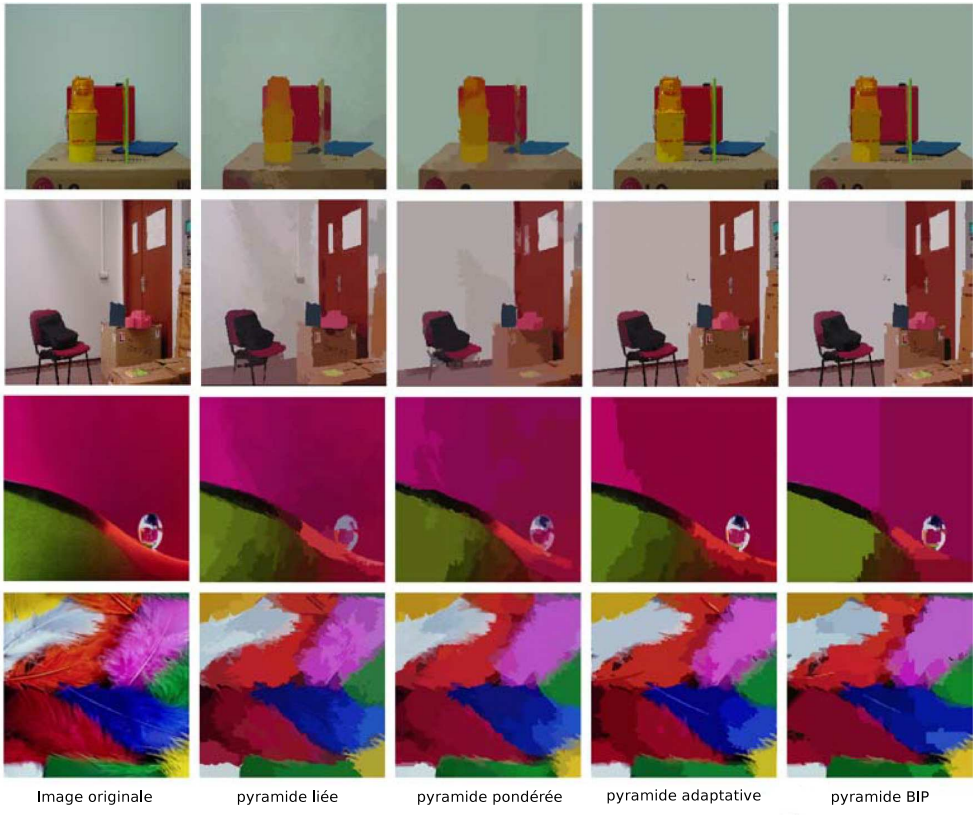


Fig. 1.7 – *Segmentations pyramidales.*

1.1.4 Segmentation par approche contour

L'extraction de contours est, dans la plupart des cas, basée sur le calcul de dérivées ou de gradients [Canny, 1986, Novak et Shafer, 1987, Alshatti et Lambert, 1993]. Ces algorithmes fournissent une carte de contours qui permet ensuite de reconstituer des régions. bien sûr il faut

"fermer" les contours afin de réaliser cette dernière étape. La fermeture des contours n'est pas l'étape la plus simple : la distinction entre le bruit et les petits objets n'est pas aisée. La figure 1.8 illustre le principe général d'une détection couleur de contours. Il s'agit encore une fois d'une utilisation dite marginale de la couleur : les trois canaux sont traités séparément et ensuite une fusion permet l'extraction finale [Zenzo, 1986, Lee et Cok, 1991].

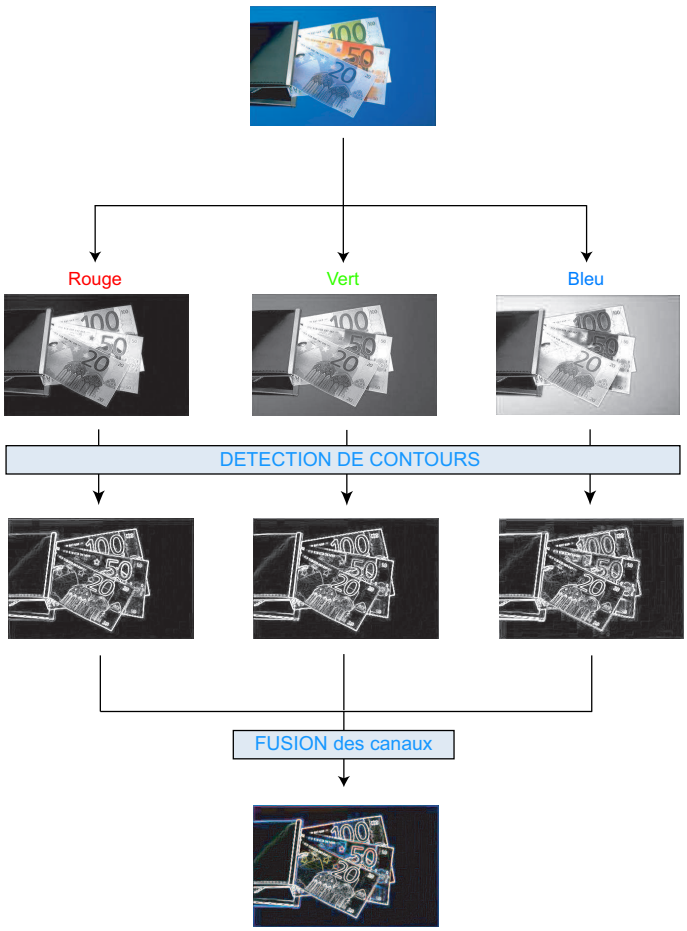


Fig. 1.8 – *Extraction couleur de contours*

Une remarque générale peut être faite au sujet des méthodes basées sur les contours : elles ne sont pas forcément adaptées pour la segmentation d'images de scènes complexes. Si ces méthodes ont montré leur efficacité dans des cas bien particuliers et seront encore d'actualité pour de nombreux problèmes, elles ont certaines limites dans le cas d'images de scènes. Le bruit, les petits objets, les niveaux de profondeurs, les dérives d'éclairages sont autant de difficultés qui perturbent la segmentation finale. Par contre, l'utilisation de détecteurs de contours afin de quantifier ces derniers, pour une utilisation en indexation, se justifie pleinement [Linhui et Kitchen, 2000].

Méthodes	Qualités et défauts
Histogrammes …	Mise en oeuvre simple Facilité d'adaptation à une problèmatique Utilisation marginale de la couleur Nécessite la présence de pics dans l'histogramme Critères de seuillage non génériques Nombre de régions issues non contrôlable
Clustering …	Rapide et efficace Utilisation vectorielle de la couleur et non marginale Nombre de régions issues non contrôlable Difficulté d'adaptation aux propriétés de l'image Nécessite un post-traitement
WaterShed …	Bien adapté pour trouver de grandes régions homogènes Retrouve les petites régions bien constrastées Parfois coûteux en temps Seuils de frontière difficiles à définir Sur-segmentation courante
Contours …	Rapide Efficace sur les images avec des constrastes élevés Problématique sur les textures Fermeture des contours délicate

Tab. 1.1 – *Propriétés des principales méthodes de segmentations*

1.1.5 Conclusion

Différentes études illustrent et critiquent les méthodes de segmentation [J.P. Cocquerez, 1995, Zhang, 1996, Lucchese et Mitra, 2001]. La conclusion généralement obtenue configure que, bien qu'il existe des méthodes de référence en terme de segmentation, aucune de celles-ci ne peut être considérée comme la meilleure dans le cadre d'images de scènes. Chaque méthode possède son lot de défauts, qui ne lui permet pas d'effectuer une segmentation efficace sur certaines classes d'images. Le tableau 1.1 résume brièvement les différentes qualités et insuffisances usuellement prêtées dans la littérature.

Au cours de notre étude nous avons mis l'accent sur différentes méthodes afin de mieux les maîtriser dans le contexte de la recherche d'images par le contenu. Dans ce contexte, un travail sur les approches basées sur la pyramide nous a permis de proposer une adaptation dont le but est d'extraire des régions grossières. En effet, les méthodes pyramidales sont des méthodes

fournissant rapidement et efficacement des régions couleurs homogènes. Il nous ainsi semblé intéressant d'approfondir cette approche et d'en proposer une adaptation fortement contrôlable. Par le fait, cette segmentation va être un point de départ à d'autres travaux que nous présenterons plus en aval dans ce manuscrit. Détaillons maintenant cette méthode pyramidale.

1.2 Méthode top-down de segmentation couleur par propagation d'étiquettes

Notre procédure de segmentation repose sur le schéma présenté figure 1.9, donnant naissance à trois étapes distinctes:

- la construction pyramidale;
- le choix des germes représentatifs des objets de l'images;
- le processus top-down d'agrégation qui engendre la segmentation définitive jusqu'à la base de la pyramide.

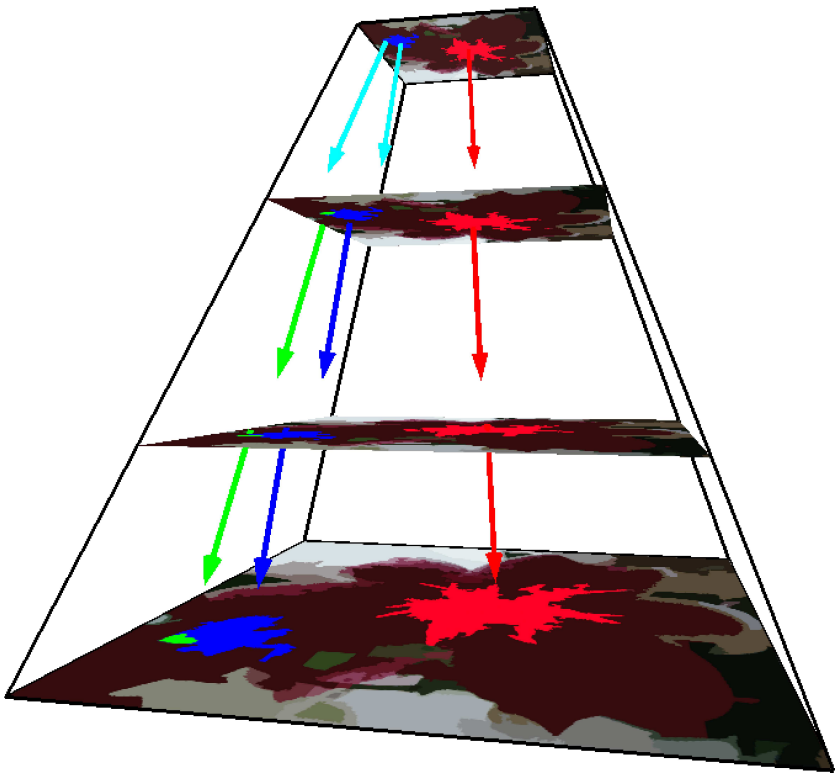


Fig. 1.9 – *Procédure de segmentation via la pyramide*

La première étape consiste à construire une pyramide gaussienne, vision multi-échelle de l'image selon le principe décrit en Annexe B. Chaque pixel d'un niveau est en fait calculé en fonction de ses pixels fils, ie ceux sensés représenter la même zone d'information mais à la taille supérieure. De l'analyse descendante de ces liens pères-fils va naître l'étape de segmentation.

1.2.1 Étape 2 : Choix des germes

Maintenant que la pyramide est construite, l'objectif est donc d'extraire les germes du processus hiérarchique. Initialement, le nombre de germes potentiel équivaut au nombre de pixels du sommet choisi. Nous avons fixé un niveau initial situé à deux niveaux en-dessous du niveau maximal de construction, engendrant par exemple des images de taille 7×11 pour des images réelles de taille 249×378. Ce choix, empirique, repose également sur une volonté d'obtention d'un nombre de régions final pas trop important. En effet, dans un contexte applicatif de recherche par le contenu dans de grandes bases de données image, un nombre conséquent de régions obtenues peut être inadapté et devenir prohibitif ([Lau et Levine, 2002]).

À partir de là, différentes approches sont possibles pour réaliser un premier étiquetage du sommet :

- *méthode G* :
  Chaque pixel devient germe et le nombre de régions initial dépend de la seule résolution du niveau sommet [Rezaee *et al.*, 2000].
- *méthode GF* :
  Chaque pixel devient germe mais une étape de fusion permet à certains germes de se regrouper en cas de forte ressemblance. En effet, il semble logique d'agréger les germes d'une région commune. Le critère de proximité peut classiquement reposer sur une distance faible calculée dans l'espace RGB ou l'espace HSV [Lew, 2001].
- *méthode GND* :
  Utilisation d'un algorithme de classification par nuées dynamiques ou k-means en fixant le nombre de germes souhaités. Afin de couvrir au maximum l'information colorimétrique de l'image, pour également améliorer la robustesse de la méthode, les germes initiaux du processus de classification occupent au mieux le nuage couleur dans le cube RGB. Pour cela, un découpage en sous-cubes est utilisé. Dans ce dernier cas, le nombre de germes est indépendant de la taille du niveau initial mais repose sur un choix empirique. Notons que cette borne supérieure peut ne pas être atteinte selon la densité colorimétrique du nuage de points.

La figure 1.10 présente ces trois possibilités sur des images de scène variées. La méthode GF est celle dont les germes sont les moins nombreux, les pixels n'étant plus agrégés que dans le cas GND. De fait, la méthode GND semble, au niveau des germes, mieux refléter l'information originale que les autres méthodes.

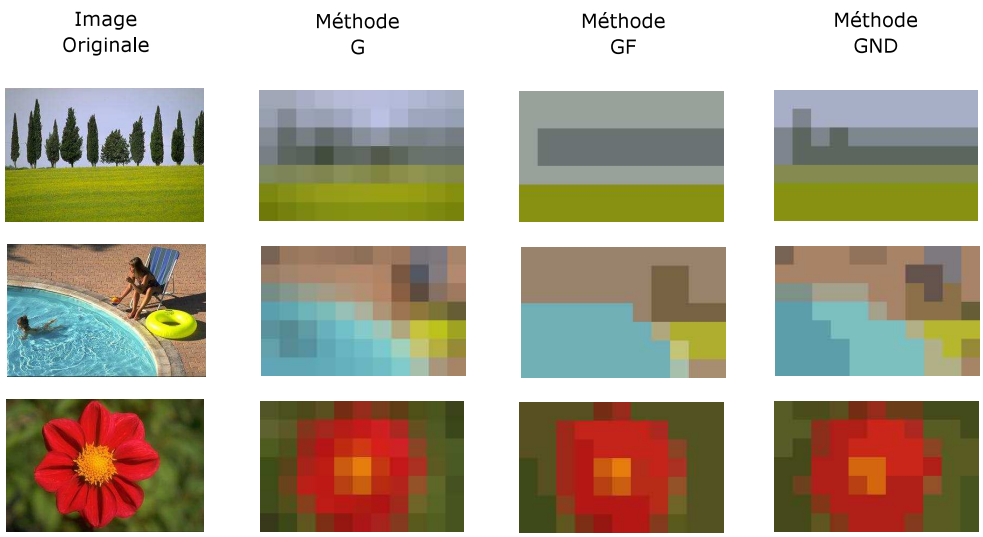


Fig. 1.10 – *Construction des germes du processus de segmentation hiérarchique.*

1.2.2 Étape 3 : Processus top-down d'agrégation

À partir du sommet, représentation grossière de l'image initiale, il est nécessaire d'emprunter les liens de descendance pour appréhender les relations spatiales et les similarités entre éléments. Les relations spatiales sont données d'une part par les notions de père et de fils présentées précédemment entre les niveaux et par les notions de voisinage classiques V_4 ou V_8 à un même niveau.

Les similarités sont dans notre cas obtenues en comparant la proximité colorimétrique entre un fils et ses pères potentiels. Dans le but d'élire le meilleur représentant, un pixel du niveau courant aura tendance à recevoir l'étiquette de son père le plus similaire. Néanmoins, dans une approche région, on va mettre à jour à chaque niveau des descripteurs pour chacune des régions définies. À partir de là, un pixel va alors recevoir l'étiquette de la région la plus proche à laquelle appartient un de ses pères. La similarité est obtenue par la distance euclidienne dans l'espace couleur $L^*a^*b^*$ entre la couleur du pixel et la couleur moyenne de la région du père[1]. De plus, les niveaux élevés représentent une version grossière de l'image. Du coup, il se peut que par moyennage des régions distinctes sur la base donnent naissance à une région commune à un niveau de faible résolution. Par conséquent, il faut donner la possibilité à une région de se scinder lors de la descente. Pour se faire, lorsqu'un fils va être jugé trop éloigné de ses pères potentiels, il va donner naissance à une nouvelle région. La notion d'éloignement, empirique, est fixée par un seuil a priori. Malgré tout, en l'absence d'un post-traitement de type fusion à chaque niveau

1. L'apparition de "fausses couleurs" par moyennage n'est pas ici véritablement problématique, ce phénomène étant intrinsèque à la construction pyramidale utilisée.

de segmentation, plutôt que de partir isolé, il va d'abord chercher à se rapprocher de ses frères - pixels voisins du même niveau - pour éventuellement former des régions communes.

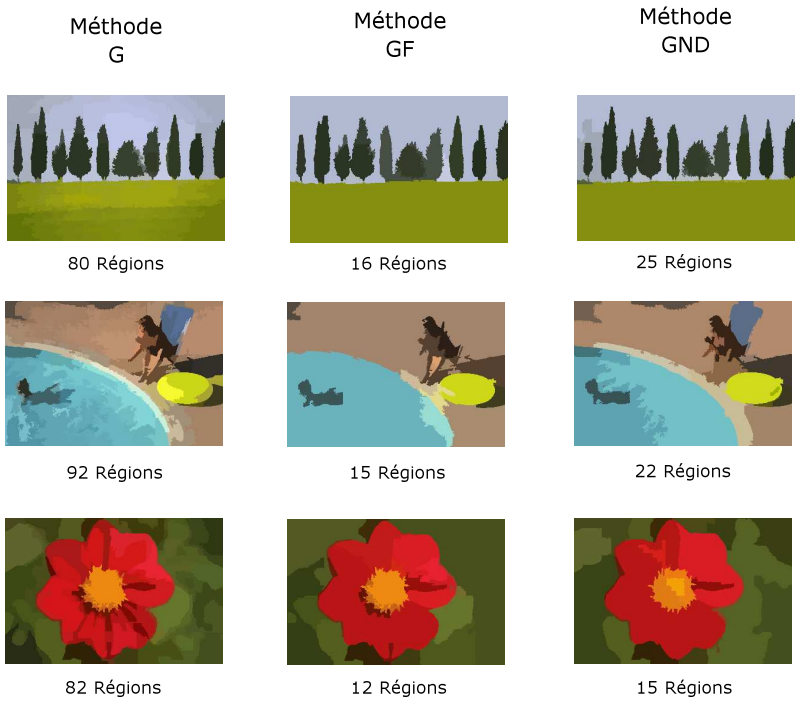


Fig. 1.11 – *Exemples de segmentations obtenues selon l'initialisation des germes.*

En définitive, présentons quelques résultats de cette segmentation sur les images dont les sommets ont déjà été présentés figure 1.10. La figure 1.11 présente ces résultats, où chaque région reçoit la valeur moyenne couleur calculée sur l'ensemble de ses pixels. Le niveau de la pyramide considéré pour la génération des germes est l'antépénultième niveau. Fait prévisible, si la génération des germes est de type un pixel égale un germe, le nombre de régions est plus important que les autres cas. Ainsi, suivant la génération de germes, il est possible d'obtenir trois types de résultats distincts:

- privilégiant des regroupement très importants (méthode GF);
- privilégiant un découpage en zones couleurs homogènes (méthode G);
- privilégiant des regroupement importants mais en conservant l'information couleur (méthode GND).

La figure 1.12 illustre la segmentation de type GND sur quelques autres images de scènes [2].

2. Les images sont de tailles diverses (de 500×400 à 1100×900) et proviennent de diverses bases généralistes où la compression est souvent un JPEG de qualité 75.

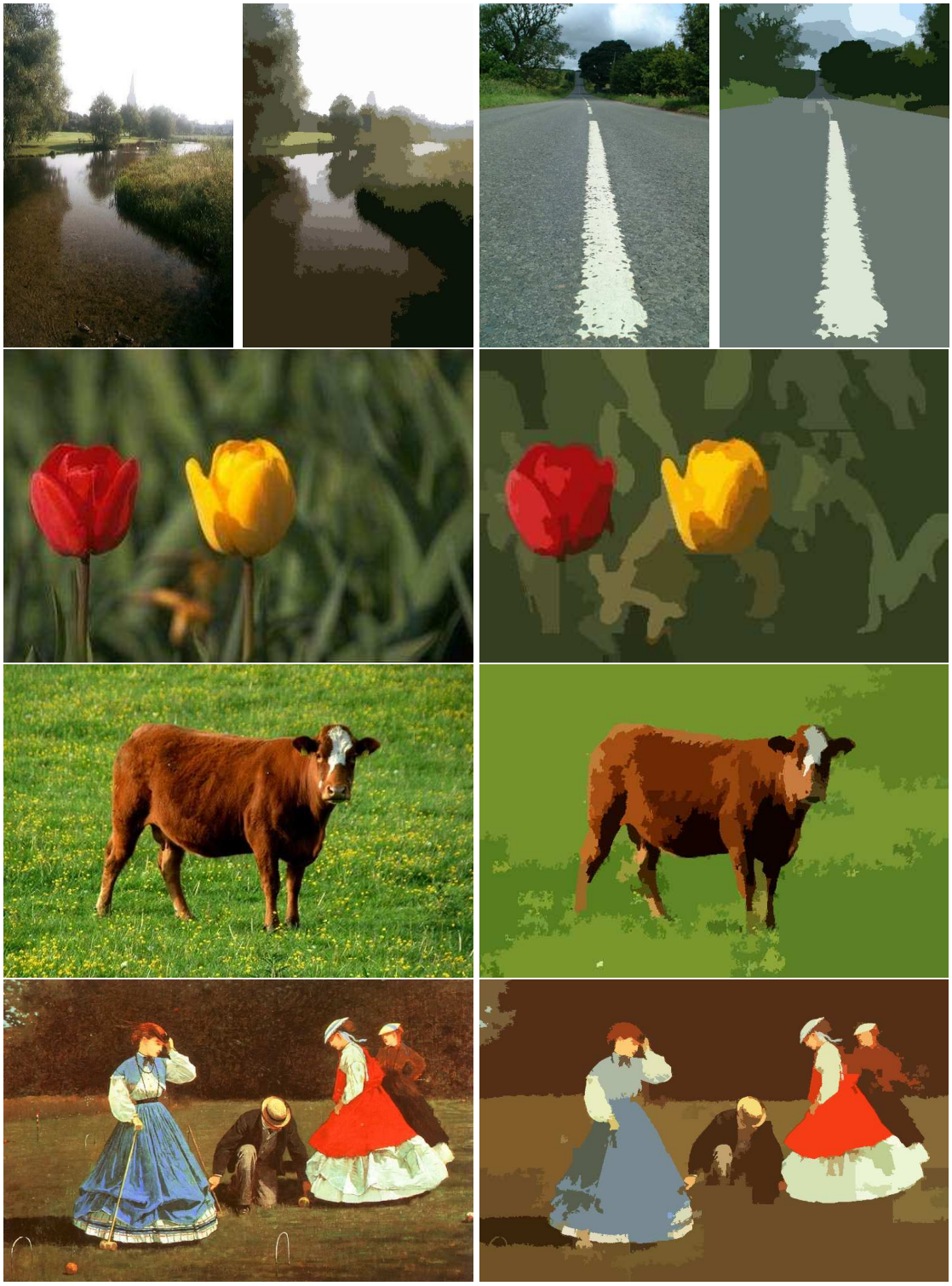


Fig. 1.12 – *Exemples de segmentations obtenues, avec une initialisation des germes de type* GND.

Méthodes de segmentation	Systèmes CBIR
Quantification en N classes	Amore [Mukherjea *et al.*, 1999] C-bird [Li *et al.*, 1999] Focus [Das *et al.*, 1997] VisualSEEk [Smith et Chang, 1996]
Histogrammes	ImageMiner [Kreyss *et al.*, 1997]
Clustering - texture/couleur/spatial	Blobworld [Carson *et al.*, 1999] MARS [Ortega *et al.*, 1997] SIMPlicity [Wang *et al.*, 2001]
Fusion - texture	CBVQ [Smith et Chang, 1995]
Contour	DrawSearch [Sciascio *et al.*, 1999]
Croissance de région	ImageRetro [Vendrig *et al.*, 1999] NETRA [Ma et Manjunath, 1999b] [Ma., 1997]
Segmentation pyramidale	Picasso [Bimbo *et al.*, 1997] *i*COBRA

Tab. 1.2 – *Méthodes de segmentations et indexation d'images*

1.3 Conclusion : segmentation et recherche d'images par le contenu

Le tableau 1.2 présente les diverses méthodes de segmentation utilisées dans des systèmes de recherche ou de navigation dans des bases d'images. Certains systèmes, comme Virage, n'utilisent pas de segmentation automatique et ne sont pas présentés ici.

Il est clair que de nombreuses approches co-existent, chaque méthode ayant son lot de réussites et de travers. Notre approche pyramidale pour des images de scènes n'échappe pas à la règle. Néanmoins une question peut alors se poser : "Comment juger telle méthode face aux autres ?" Une réponse directe pourrait être, comme tout ce qui touche à l'indexation d'images, qu'en matière de segmentation tout est relatif. De plus, dans la littérature, la plupart des évaluations objectives des méthodes de segmentation reposent en fait sur la capacité des différentes approches à respecter une segmentation dite de référence [Zhang, 1996, Zhang, 1997, Borsotti *et al.*, 1998].

Aucun protocole de test ou de calcul de mesures ne permet de qualifier une méthode de segmentation face à un problème de recherche d'images par le contenu. Nous même, nous avons présenté des exemples de segmentation qui semblent visuellement correctes. Néanmoins nous ne pouvons appuyer notre discours par des chiffres qui montreraient l'efficicacité ou non de la méthode de manière indiscutable. Pour répondre à ce manque crucial d'appréciation des méthodes de segmentation dans notre contexte bien précis, nous proposons donc un protocole d'évalution

objectif, que nous allons tout d'abord détailler puis mettre en oeuvre sur quelques algorithmes de référence.

Chapitre 2

ÉVALUATION DES MÉTHODES DE SEGMENTATIONS

Sommaire

Juger une méthode de segmentation n'est pas une volonté récente. Néanmoins il n'existe pas d'évaluation objective, dans un contexte indexation d'images, de cette étape incontournable dans l'extraction de connaissances. Cette étude introduit alors un protocole d'évaluation objective des méthodes de segmentation et des mesures spécifiques permettant de qualifier leur stabilité. Ensuite, quatre méthodes classiques de segmentation: Meanshift, Clustering par nuées dynamiques, WaterShed et finalement l'approche pyramidale présentée au chapitre précédent seront, dans un deuxième temps, évaluées via ce protocole.

2.1 Introduction

L'objectif de cette étude [1] est de proposer un protocole d'évaluation des méthodes de segmentation. Nous voulons mesurer la confiance que l'on peut accorder à la méthode elle-même, sans insérer dans l'évaluation la phase de recherche de similarité propremente dite. Nous présenterons ensuite une mise en œuvre de ce protocole dans le cas de quatre méthodes de segmentation classiques, couvrant les principales familles depuis l'analyse du nuage colorimétrique jusqu'à l'approche région. De plus, nous voulons quantifier la robustesse de cette étape selon diverses variations classiques venant perturber l'information de l'image, depuis un changement d'illuminant jusqu'aux effets de la compression.

2.2 Protocole d'évaluation objectif

2.2.1 Définition du protocole

Dans la plupart des travaux sur l'évaluation des méthodes de segmentation, les résultats sont jugés par rapport à une segmentation de référence habituellement obtenue manuellement. La méthode sera jugée d'autant meilleure que la segmentation se rapprochera de celle espérée ([Zhang, 1996], [Zhang, 1997], [Borsotti *et al.*, 1998], [Correia et Pereira, 2002]). Néanmoins, cette vision du problème suppose intrinsèquement l'existence d'une segmentation parfaite, ou encore non sujette à caution. Dans un contexte généraliste de recherche par le contenu, il semble de toute évidence plus critique de suivre un tel cheminement, notamment par l'absence de segmentation de référence. En effet, le côté subjectif des résultats attendus ne peut être satisfaisant si l'on désire une évaluation objective. En définitive, la limite provient encore une fois de l'absence de segmentation sémantique capable d'extraire la bonne information et ceci de façon totalement automatique pour toute recherche et toute image.

Cependant, il est indispensable, pour pouvoir assumer une tâche de reconnaissance, de certifier l'extraction automatique de certains objets particuliers, d'une image à l'autre, et ceci indépendamment du contexte dans lequel ils sont plongés. Si l'on caractérise par exemple un pâquerette comme étant un rond jaune entourée d'une zone blanche, la reconnaissance de cette fleur implique que le segmentation fournisse distinctement ces deux zones. Il est ainsi préférable de donner crédit à une méthode stable quel que soit le contexte environnant dans lequel apparaît l'objet de référence à isoler. En définitive, notre protocole d'évaluation repose sur la sélection d'objets divers plongés aléatoirement au sein d'une base généraliste. La robustesse et la qualité accordées à telle ou telle méthode de segmentation seront fonction de la stabilité obtenue entre la segmentation de référence et la segmentation automatique, selon le schéma présenté figure 2.1.

1. Ce travail à fait l'objet d'une publication [Da Rugna et Konik, 2004b].

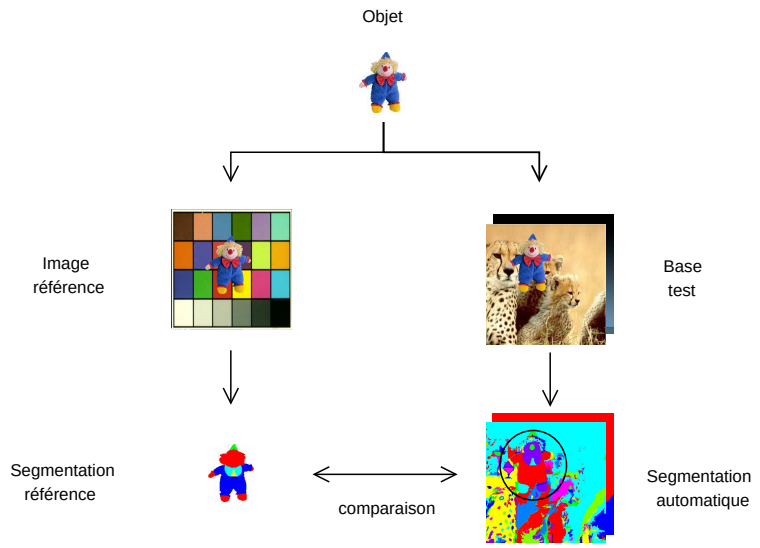


Fig. 2.1 – *Diagramme du protocole d'évaluation.*

Un objet est dans ce protocole plongé d'une part dans un fond spécifique, et d'autre part dans des images de scènes. Ce fond spécifique permet ainsi de construire une segmentation de référence de l'objet. D'autre part, le plongement dans une image de scène permet de construire une segmentation de l'objet que l'on compare ainsi à la segmentation de référence. La localisation de l'objet, dans l'image de scène, est réalisée simplement en conservant au moment du plongement la position exacte de l'objet, via le rectangle englobant. En effet, on ne désire pas évaluer la capacité à retrouver l'objet au milieu de son contexte mais la capacité à segmenter de manière stable ce dernier.

2.2.1.1 Base d'objets

Dans ce processus, il convient de choisir des objets de référence. Comme l'indiquent les auteurs [Muller *et al.*, 2001], une des problématiques de l'évaluation objective des méthodes de recherche par le contenu repose, entre-autres, sur la non-exhaustivité de toute base choisie comme référence. Néanmoins, pour ce protocole nous avons sélectionné une série d'objets formant le panel le plus représentatif possible de l'indexation d'images. La sélection s'est faite notamment suivant les différentes catégories suivantes :

- objets mono-colorés ;
- objets multi-colorés ;
- objets mono-texturés ;
- objets multi-texturés.

Habituellement, ces objets de référence forment visuellement un tout mais il est tout à fait possible que chacun d'entre eux donne naissance à plusieurs régions par telle ou telle approche, sans que cela soit pénalisant. Nous ne nous situons pas ici dans une optique de regroupement des différents blobs pour former un tout sémantique [Luo et Guo, 2003]. Les auteurs différencient les regroupements d'ordre NPG ("non-purposive grouping"), basés sur les concepts généraux d'une bonne segmentation, et d'ordre PG, ("purposive grouping"), où certains modèles spécifiques peuvent contraindre certaines étapes pour reconstituer des objets plus proches de la réalité de l'utilisateur lambda. Nous nous situons ici encore une fois dans une approche de non-corrélation entre régions extraites et objets [Fuertes *et al.*, 2001], et nous conserverons les régions extraites sans post-traitement, soit telles qu'elles sont obtenues directement par la méthode.

Plus précisément, la réalité ne repose pas sur cette classification simpliste et chacune de ces catégories peuvent en fait ou non cœxister dans un échantillon retenu. Par exemple, sur la figure 2.2 présentant certains objets de référence, la rosace entre dans une catégorie multi-colorée et mono-texturée alors que la poupée est aussi multi-colorée mais par contre multi-texturée. De plus, certains objets ont été choisis pour leur forme spécifique, qu'elle soit allongée (le roi) ou non pleine (la pièce).

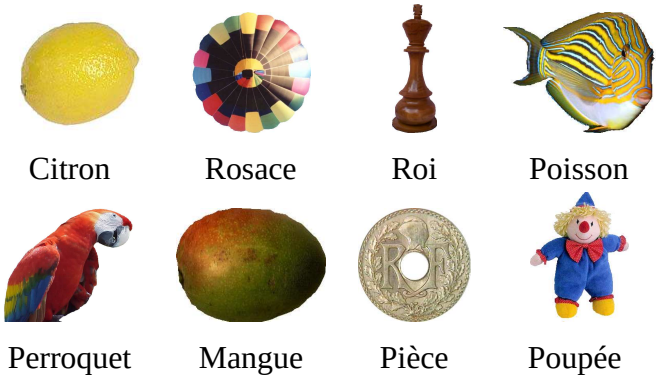


Fig. 2.2 – *Objets de références*

2.2.1.2 Segmentations de référence

Comme précédemment évoqué, chaque objet peut donner naissance à une segmentation en multiples régions. Ainsi il est nécessaire d'en obtenir une au préalable, que nous appellerons segmentation de référence. Pour cela, il semble plus logique de ne pas considérer uniquement l'objet mais de la plonger au coeur d'un fond standard commun à tous les objets. Il est en effet exclu d'effectuer l'étape de segmentation sur l'objet isolé, en faisant abstraction du contexte pour une raison évidente: la segmentation de l'objet sans le fond, donc sans le contexte, est trop fortement biaisée par rapport à une segmentation une fois l'objet plongé dans une image de scène.

Il est clair que ce choix ne peut pas être anodin tant l'importance de ce fond va contraindre les résultats de ces segmentations initiales. Du coup, afin de ne biaiser aucune méthode, nous avons opté pour la mise en place de plusieurs fonds (présentés figure 2.3):

- fond "bruit" constitué d'un bruit gaussien couleur dense ;
- fond color "chart" constitué d'une planche MacBeth de référence ;
- fond homogène "blanc".

L'idée est d'utiliser ces différents fonds pour montrer qu'il n'existe pas une unique segmentation de référence mais que la robustesse globale d'une méthode n'est pas liée au fond considéré.

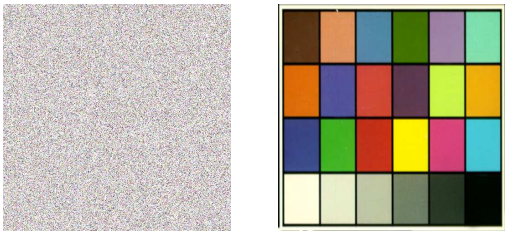


Fig. 2.3 – *Les fonds non-blanc de référence*

Plus précisément, les deux premiers fonds reposent sur une discrétisation de l'espace couleur presque identique mais sont situés aux deux extrémités d'un continuum caractéristique de l'agencement spatial (du pur aléatoire vers le tout organisé). Nous disposons d'abord d'un fond sans organisation spatiale et sans contours marqués, autant dire sans objet véritable. Le deuxième fond est organisé avec des régions homogènes bien marquées présentant des contours nets. Ainsi, chaque famille de méthodes de segmentation devrait trouver un fond "naturellement" plus propice à engendrer des résultats satisfaisants.

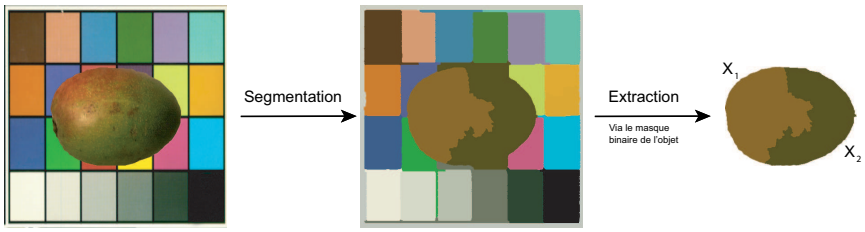


Fig. 2.4 – *Création de la segmentation de référence*

En définitive, comme l'illustre la figure 2.4, la segmentation de référence que nous considérons est l'intersection de la segmentation issue du plongement de l'objet dans un fond et du masque binaire de l'objet. Il aurait été aussi possible de sélectionner toutes les régions recouvertes totalement ou non par l'objet. Néanmoins, la nature très artificielle de nos fonds de référence ne justifierait pas cette approche. De plus, dans la grande majorité des cas, la segmentation

ne "bave pas" sur le fond et aucune région ne recouvre en même temps le fond et l'objet. On notera alors $X = \bigcup_{i=1}^{m} X_i$ une segmentation référence de l'objet. X correspond donc exactement à l'ensemble des pixels de l'objet. Sur la figure 2.4, la mangue est par exemple divisée en deux régions X_1 et X_2.

2.2.1.3 Plongement dans des images de scènes

En définitive, il est nécessaire de plonger artificiellement chacun de nos objets de référence dans des images de scènes, comme illustré par la figure 2.5. Pour cela, nous avons retenu une collection composée de 3000 images très diverses en terme de complexité visuelle. Afin de ne pas biaiser les résultats, les images de scènes sont de tailles semblables à celle des fonds. Ainsi l'objet représente, autant d'un point de vue spatial que colorimétrique, la même part d'information dans l'image. La figure 2.6 présente quelques exemples tirés de notre collection d'images. La position du rectangle englobant l'objet est ici conservée afin de permettre l'étape de localisation post-segmentation.

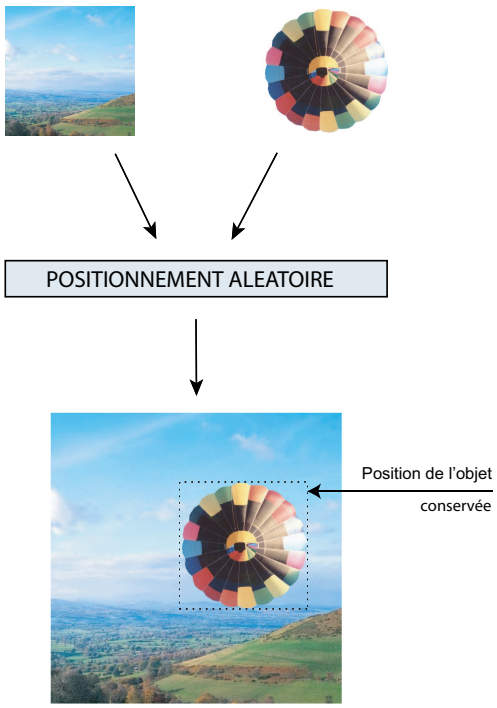


Fig. 2.5 – *Plongement des objets dans les images de scènes*

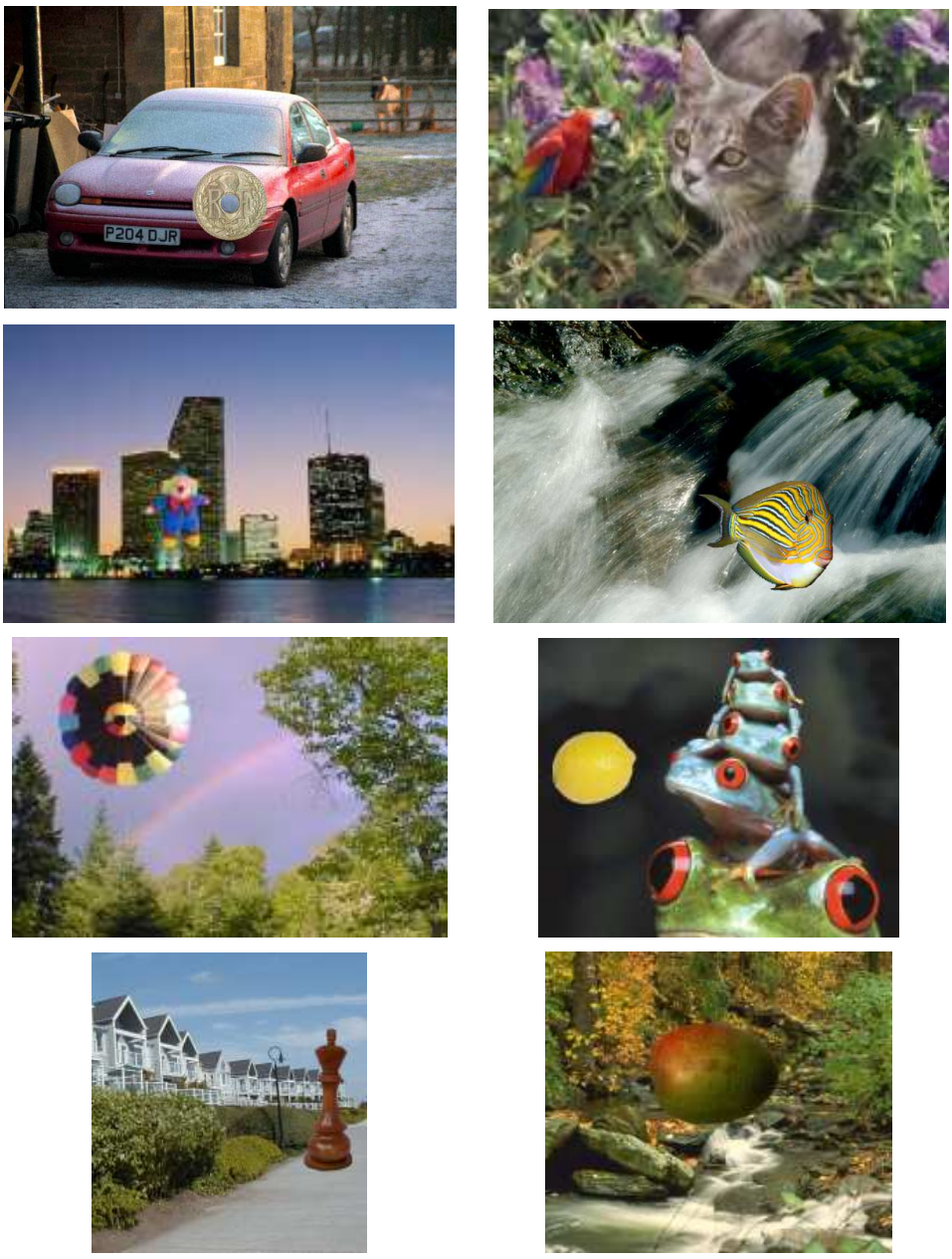


Fig. 2.6 – *Quelques images de la base test*

2.2.2 Descripteurs

À partir de là, chaque image va être segmentée, sans utiliser en quoi que ce soit la connaissance de la position de l'objet. Il s'agit donc alors de comparer la segmentation obtenue de l'objet avec celle dite de référence. Le problème repose donc maintenant sur la définition de descripteurs de ressemblance entre deux segmentations. En écartant pour l'instant le fait qu'il est plus raisonnable d'attendre un nombre d'objets réduit [Lau et Levine, 2002], notre problématique est ainsi de modéliser par des mesures objectives la potentialité de retrouver l'objet dans l'image. Nous l'avons vu précédemment, on retrouve dans la majorité des approches régions l'aspect couleur et/ou l'aspect spatial. Bien sûr, différentes voies ont été explorées, la littérature [Borsotti *et al.*, 1998, Shaffrey *et al.*, 2002, Roman-Roldan *et al.*, 2001] offre diverses mesures permettant de juger les segmentations. Nous avons aussi approfondi des mesures simples, comme un calcul d'un *RMSE*, toujours sujet à caution, entre les deux segmentations pour mesurer la concordance couleur par exemple. Au final, nous avons retenu trois mesures, bien distinctes, dont l'objectif est de présager de l'efficacité d'une recherche de similarité par régions.

Tout d'abord, d'un point de vue morphométrique, une segmentation peut être considérée comme correcte si l'objet est bien détaché du fond et si le découpage n'a pas trop varié par rapport à la segmentation de référence. En effet, même si les objets sont découpés initialement en plusieurs zones, encore faut-il qu'elles soient physiquement positionnées de façon stable afin de ne pas trop perturber l'étape de recherche de similarité en aval. L'information de positionnement relatif entre les différentes parties est en effet grandement utilisée pour assumer la tâche de reconnaissance. Nous allons donc utiliser deux mesures : l'une mesurant la pertinence des régions obtenues, au sens où elles ne doivent pas trop déborder sur le contexte environnant, ce qui risquerait de limiter l'étape de recherche future, l'autre mesurant leur stabilité, notamment pour ne pas que les vecteurs de caractéristiques extraits ne soient totalement bouleversés.

De plus, afin de mesurer l'adéquation du nuage couleur de l'objet initial avec le nuage couleur des régions segmentées, nous avons introduit une mesure basée sur un histogramme couleur. L'objectif est de pondérer la possibilité pour un objet de se fondre de façon cohérente avec son fond. En effet, si un objet jaune est plongé au sein d'un fond jaune, le fait que la segmentation soit moins bonne est visuellement acceptable. Du coup, nous cherchons à pondérer ce cas de figure à l'aide de cette mesure [Colombo *et al.*, 1997, Duda *et al.*, 2001], basée sur des différences colorimétriques. Ainsi, nous évaluerons la capacité d'une méthode de segmentation à être employée dans le cadre d'une recherche basée sur des différences colorimétriques.

2.2.2.1 Coefficient de mélange

Le premier descripteur, dénommé *coefficient de mélange*, mesure le pourcentage de pixels de l'objet mélangé au fond. Une trop grande dispersion sur les régions voisines risque en effet

de limiter la reconnaissance de l'objet. Nous devons néanmoins garder une certaine souplesse, et si l'objet sort de moins de $x\%$ du support idéal, il sera considéré comme valide (x étant bien évidemment assez faible).

Soit $Y = \bigcup_{j=1}^{n} Y_j$ une segmentation de l'objet dans l'image de scène. $\bar{X}$ représente la partie complémentaire de l'objet, ie l'ensemble des pixels de l'image qui ne font pas parti de l'objet. Le coefficient de mélange a pour valeur :

$$CM = \frac{\sum_{j=1}^{n} (Card(Y_j \cap X) \times \delta_j)}{Card(X)} \tag{2.1}$$

avec

$$\delta_j = \begin{cases} 1 \text{ si } \dfrac{Card(Y_j \cap \bar{X})}{Card(Y_j)} \geq x \\ 0 \text{ sinon} \end{cases} \tag{2.2}$$

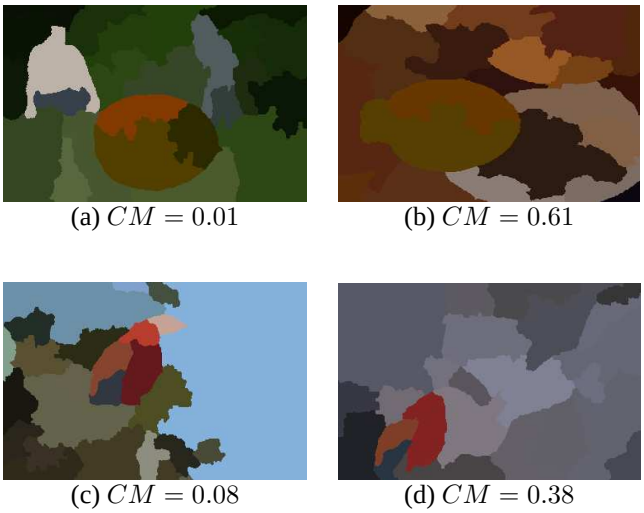
La figure 2.7 illustre deux exemples pour l'objet "mangue" d'une part et pour l'objet "perroquet" d'autre part, où la valeur affectée en chaque pixel est la valeur moyenne calculée pour la région à laquelle il appartient. Plus le coefficient de mélange est fort et plus les régions détectées par la segmentation de l'objet sont noyées dans son fond environnant, ce qui rend d'autant plus difficile sa reconnaissance.

Notons que deux hypothèses principales rendent le "bavage" possible :

- La non différenciation. Si la couleur du fond touchant l'objet est similaire à l'objet le long de la frontière, alors il est très difficile pour un algorithme de segmentation de différencier l'objet de son fond. Ainsi une partie de l'objet est amalgamée avec le reste de l'image.
- La non représentativité de certaines zones. Certaines zones couleur de certains objets, comme par exemple le jaune sur l'aile du perroquet, ne forment pas un ensemble couleur étendu. Ainsi, dans le cas d'un algorithme par clustering par exemple, cette zone peut soit représenter un cluster, soit être confondue dans un autre cluster. Au final, suivant les deux possibilités, la zone sera fusionnée, ou non, avec une autre zone de l'image, qui pourrait être le fond de celle-ci.

2.2.2.2 Mesure de Vinet

Pour le second descripteur, sensé mesurer la concordance spatiale entre les régions extraites, la littérature offre de nombreuses approches classiques reposant sur une matrice de confusion où chaque entrée C_{ij} représente le nombre de pixels de classe j classifiés en tant que classe i par l'algorithme de segmentation. Suivant la démarche adoptée en [Cocquerez et Philipp, 1995], nous



(a) $CM = 0.01$ (b) $CM = 0.61$

(c) $CM = 0.08$ (d) $CM = 0.38$

Fig. 2.7 – *Exemples de coefficients de mélange obtenus avec* $x = 5\%$

allons utiliser la *mesure de Vinet* reposant sur la détermination des couples de régions assurant un recouvrement maximum entre les deux segmentations. La caractérisation de la dissimilarité est induite par la proportion de pixels ne participant pas à ce recouvrement.

Posant N comme le nombre de pixels total de l'objet initial considéré, le principe est le suivant :

- définition de la table de superposition T par :

$$T(i,j) = Card(X_i \cap Y_j) \quad \begin{matrix} \forall i & = 1..m \\ \forall j & = 1..n \end{matrix}$$

- recherche du couple de régions (X_{i_1},Y_{j_1}) de recouvrement maximal tel que :

$$T(i_1,j_1) \geq T(i,j) \quad \forall i,j$$

  On note $c_1 = T(i_1,j_1)$.

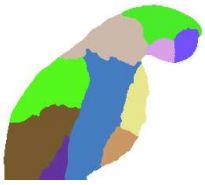
- itération jusqu'à obtenir (X_{i_k},Y_{i_k}), avec $k = \min(m,n)$, tel que :

$$T(i_k,j_k) \geq T(i,j) \quad \begin{matrix} \forall i & \neq i_1,i_2,...,i_{k-1} \\ \forall j & \neq j_1,j_2,...,j_{k-1} \end{matrix}$$

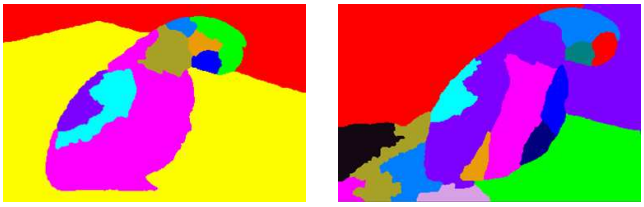
- calcul de la mesure de dissimilarité par :

$$mV = \frac{N - \sum_{i=1}^{k} c_i}{N} \tag{2.3}$$

La figure 2.8 illustre deux exemples de valeurs obtenues pour l'objet "perroquet" dont la segmentation de référence est préalablement donnée. Plus la distance est faible plus la segmentation de l'objet est proche de l'originale, ce qui facilitera d'autant la recherche de similarité avale.



Segmentation de référence



$V = 0.7$ $V = 0.2$

Fig. 2.8 – *Exemples de mesures de Vinet obtenues.*

2.2.2.3 Mesure histogramme couleur

Décrivons maintenant la mesure couleur nous permettant de juger de l'adéquation colorimétrique entre deux groupes de régions.

X étant la segmentation de référence et Y la segmentation de l'objet dans l'image de scène, nous générons deux histogrammes couleur H_X et H_Y. Chaque région segmentée devient un élément de l'histogramme, représentée par son nombre d'éléments et sa couleur moyenne. Pour que la comparaison puisse avoir un sens, quelle que soit la variation subie par l'objet, la couleur moyenne de référence est obtenue en utilisant la couleur de l'objet telle qu'elle est au moment du plongement dans l'image. Quant à la segmentation de l'objet dans l'image de scène, on considère toutes les régions non tronquées recouvrant l'objet comme candidates à l'élaboration de l'histogramme.

H_X et H_Y possèdent les propriétés suivantes :

$$\begin{aligned} \sum_{espacecouleur} H_X(couleur) &= N \\ \sum_{espacecouleur} H_Y(couleur) &\geq N \end{aligned}$$

Dans la majorité des cas, H_Y recouvre un nombre de pixels notablement plus grand que H_X. Le but est de mesurer, sans notion de positionnement spatial, l'adéquation entre H_X et H_Y. On désire mesurer si la disposition colorimétrique dans H_X est similaire à la disposition colorimétrique dans H_Y et si chacune des régions de H_X est assimilable à une région de H_Y.

Introduisons tout d'abord une notion de mesure couleur pondérée par la taille entre deux régions R_1, de couleur $c1$ et R_2, de couleur $c2$, appartenant respectivement à la segmentation X et à la segmentation Y. La couleur d'une région est obtenue comme étant la moyenne des pixels la composant.

$$d_{ponderee}\,(R_1,R_2) = \frac{1}{1+\frac{1}{offset}} \times \left(d_{couleur}(c1,c2) + \frac{\|H_X(c1) - H_Y(c2)\|}{offset} \right) \tag{2.4}$$

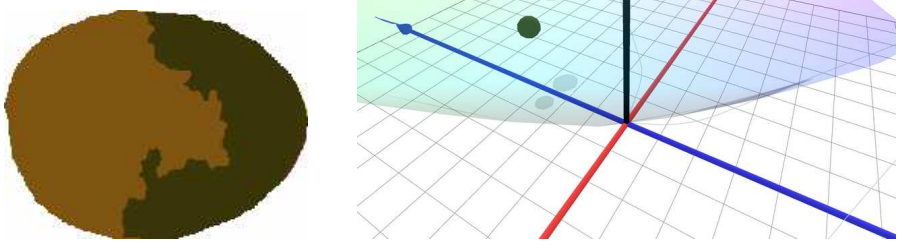
$d_{couleur}$ est la distance couleur euclidienne calculée dans $L^*a^*b^*$.
$offset$ est un coefficient pondérateur ajustant précisément le poids que l'on désire mettre à la distance entre les tailles de régions. Il est logique que l'on veuille des couples de régions de même couleur et de même taille, mais ici, on désire véritablement accentuer l'aspect couleur face à l'aspect surface.

Ensuite une étape de corrélation entre les deux histogrammes permet de calculer la mesure couleur entre les deux segmentations. Plus précisément, un appariement classique entre régions permet de générer un ensemble de couples (R_i,R_j), que l'on nomme K, une région de taille importante pouvant intervenir dans plusieurs couples par un appariement de type "1 à N".

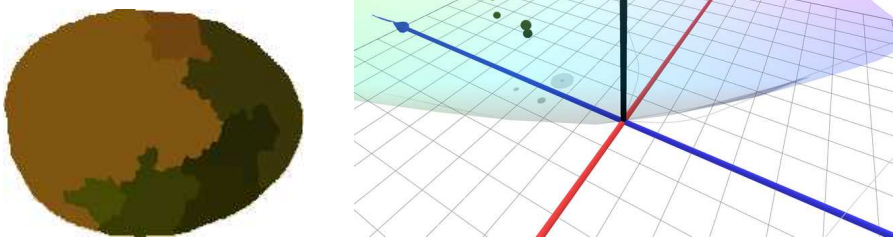
Finalement, la distance Histogramme Couleur mHC est obtenue comme suit :

$$mHC = \frac{\sum_{(i,j)\in K} d_{ponderee}\,(R_i,R_j)}{\|K\|} \tag{2.5}$$

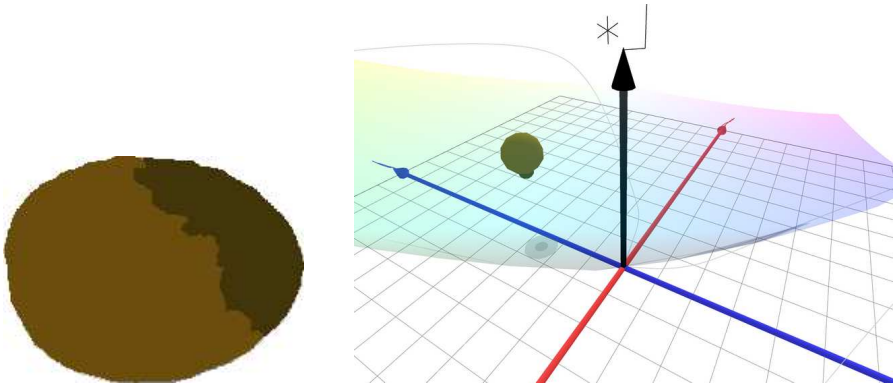
La figure 2.9 montre un exemple de mesure Histogramme Couleur sur l'objet mangue : plus le coefficient est fort, moins le nuage couleur correspondant est similaire au nuage de référence. Inversement, plus la distance est faible et plus les régions mises en jeu dans le calcul sont voisines colorimétriquement. Sur cet exemple, la région de plus grande taille a en effet beaucoup varié par rapport à sa jumelle de référence alors que dans l'autre cas, une des régions a eu simplement tendance à se morceler, via un phénomène de sur-segmentation.



Segmentation de référence — Histogramme, espace $L^*a^*b^*$



$mHC = 0.15$



$mHC = 0.6$

Fig. 2.9 – *Exemples de mesures Histogramme Couleur obtenues*

Ainsi cette mesure permet de qualifier la stabilité colorimétrique de la méthode. Car, si le partitionnement spatial change, cela ne signifie pas forcément que les régions ont évolué colorimétriquement. La sur-segmentation notamment aboutit à un recouvrement faible mais il est pourtant évident que l'information couleur est encore présente de façon quasi identique.

2.3 Les méthodes de segmentation testées

Afin d'illustrer ce protocole par un exemple réel, il est nécessaire d'introduire des méthodes de segmentations significatives. En nous basant sur notre étude du chapitre 1 et plus précisément sur le tableau 1.2 de ce dernier, nous avons opté pour quatre méthodes :

- Approche via l'algorithme "meanshift"
  Cette méthode fournit des résultats reconnus par la communauté et forme ainsi un bon étalon pour les autres méthodes. Rappelons cependant la tendance, pas forcément adaptée à notre contexte, de cette méthode à générer un nombre important de régions.
- Approche morphologique "watershed"
  L'algorithme watershed [de Andrade *et al.*, 1999] choisi permet de contrôler le nombre de bassins versants résultants. Ainsi le problème lié à une sur-segmentation ou une sous-segmentation est a priori limité.
- Approche "clustering"
  L'approche par clustering, comme nous l'avons vu précédemment, est utilisée dans de nombreux moteurs de recherche. Nous avons ainsi utilisé une méthode classique de clustering par nuées dynamiques relayée par une étape de fusion de régions[Liew *et al.*, 2001].
- Approche multirésolution "pyramide"
  L'approche pyramidale, et plus précisément la segmentation présentée précédemment, constitue la quatrième et dernière méthode de cette évaluation. La génération de germes retenue est la génération nommée GND(germes nuées dynamiques).

Il nous faut noter ici que le réglage des seuils mis en jeu a été défini une fois pour toutes, notre objectif étant de fournir les résultats les plus probants possibles. Nous avons ainsi, sur un jeu restreint d'images, effectué un certain nombre d'essais et de réglages afin d'obtenir les meilleurs résultats.

Nous présenterons pour chaque méthode retenue un exemple de segmentation de référence sur différents objets à la figure 2.1, en utilisant un plongement sur les trois fonds possibles afin de se convaincre, dans un premier temps, de leur influence.

Les figures 2.10 et 2.11 montrent les segmentations de référence pour les objets *poupée* et *mangue*. Ainsi, quelques-unes des caractéristiques principales des différentes méthodes choisies

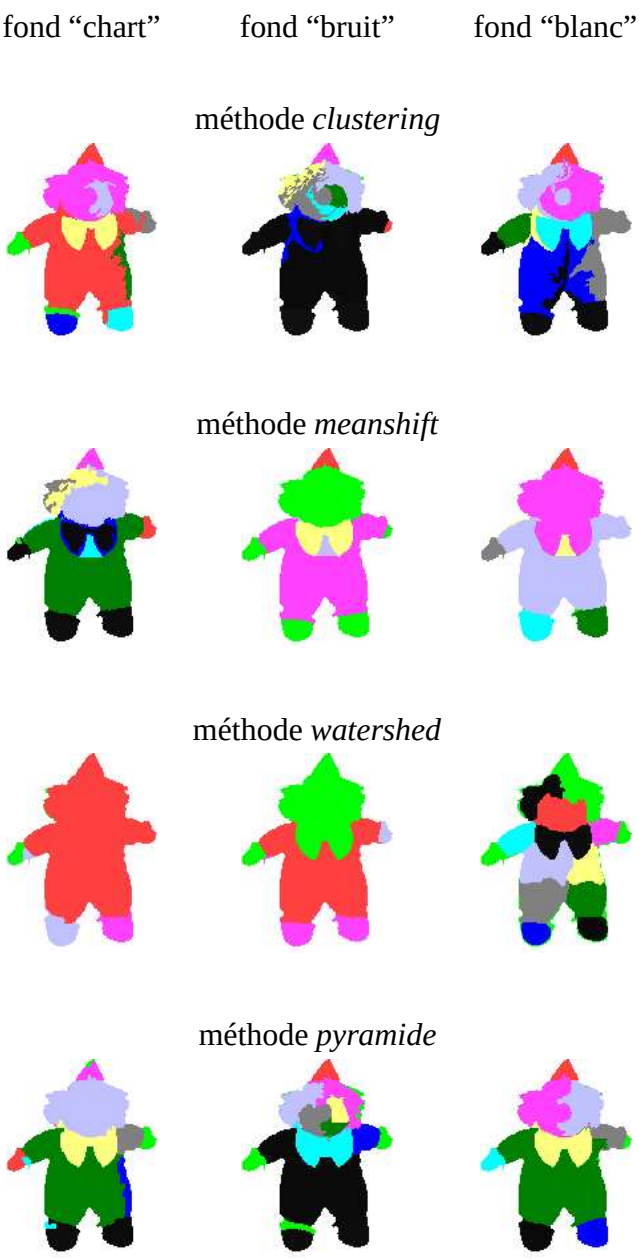


Fig. 2.10 – *Exemples de segmentations de référence sur l’objet “poupée” en fonction du fond.*

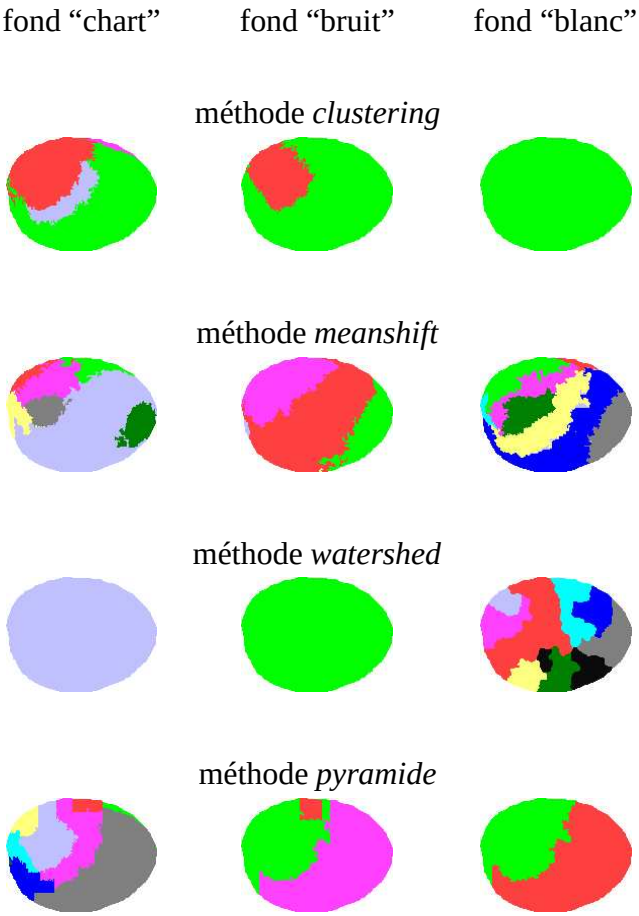


Fig. 2.11 – *Exemples de segmentations de référence sur l’objet “mangue” en fonction du fond.*

apparaissent en fonction des trois fonds . Première remarque : le fond semble influencer considérablement la segmentation de référence. Le fond blanc génère un plus grand nombre de régions, ce qui est conforme au fait que le fond homogène n'est pas segmenté en sous parties comme dans le cas des autres fonds. Sur ces exemples, il semble néanmoins difficile d'extraire une méthode a priori meilleure.

2.4 Évaluation objective

Cette évaluation objective se déroule en quatre étapes. Tout d'abord, afin de montrer que le choix des descripteurs et des différents fonds est justifié, nous étudierons la corrélation entre les descripteurs d'une part, et entre les fonds d'autre part. Une deuxième étape se propose, par rapport à chaque objet, de calculer les valeurs des trois descripteurs pour chaque méthode de segmentation. La troisième étape considère l'influence des fonds et le biais qu'ils peuvent générer. Finalement, nous montrerons le comportement des méthodes de segmentation face à diverses évolutions que subissent l'objet ou l'image elle-même.

Cette étude fournit un très grand nombre de chiffres et résultats. Cependant, afin de ne pas trop noyer l'information, nous limiterons les résultats exposés dans certains cas. En particulier, nous ne montrerons pas les résultats pour tous les fonds. Le fond sélectionné influence bien sûr les résultats obtenus et pourrait a priori biaiser les conclusions que l'on en déduit. Malgré tout, les résultats qui illustrent notre propos montrent une tendance générale commune à tous les fonds et nos conclusions sont issues de l'étude dans son ensemble.

2.4.1 Corrélation entre les descripteurs et entre les fonds

MeanShift	CM	mV	mHC
CM	1.00	0.12	0.23
mV	0.12	1.00	0.23
mHC	0.23	0.23	1.00

Clustering	CM	mV	mHC
CM	1.00	0.04	0.08
mV	0.04	1.00	0.08
mHC	0.08	0.08	1.00

WaterShed	CM	mV	mHC
CM	1.00	0.03	0.10
mV	0.03	1.00	0.04
mHC	0.10	0.04	1.00

Pyramide	CM	mV	mHC
CM	1.00	0.02	0.12
mV	0.02	1.00	0.16
mHC	0.12	0.16	1.00

Tab. 2.1 – *Corrélation entre les descripteurs.*

mV	“chart”	“bruit”	“blanc”
“chart”	1	0.72	0.25
“bruit”	0.72	1	0.24
“blanc”	0.25	0.24	1

mHC	“chart”	“bruit”	“blanc”
“chart”	1	0.74	0.72
“bruit”	0.74	1	0.75
“blanc”	0.72	0.75	1

Tab. 2.2 – *Corrélation entre les fonds.*

Vérifier une certaine indépendance de chaque descripteur revient à s’assurer que chacun d’entre eux porte bien une information distincte des autres. Pour cela, les tableaux 2.1 donnent les taux de corrélation entre chaque descripteur, calculés sur l’ensemble des objets et des fonds, pour chaque méthode de segmentation. Il apparait clairement que les trois descripteurs proposés sont non-corrélés, puisque l’intervalle de variation de la corrélation 2 à 2 est $[0.02,0.23]$. Ainsi, le choix de chacun d’entre eux se justifie pleinement et semble bien apporter une information qui lui est propre pour orienter le choix de la méthode de segmentation en fonction du scénario envisagé. Avant toute chose, reprenons rapidement ce qu’expriment ces descripteurs individuellement :

- Le *coefficient de mélange*, noté CM, mesure la capacité de la méthode à bien isoler l’objet de son fond environnant.
- La *mesure de Vinet*, notée mV, mesure la stabilité de la partition spatiale de l’objet originale.
- La mesure *histogramme couleur*, notée mHC, mesure la stabilité du nuage colorimétrique de l’objet segmenté.

À titre de remarque, signalons que pour le cas particulier du *coefficient de mélange*, nous avons retenu le seuil de 5%, sachant qu’une étude pour des valeurs allant de 5% à 20% ne montre pas d’écart significatif.

Dans un second temps, il est nécessaire de caractériser l’influence du fond sur les seuls descripteurs mV et mHC, puisque ce choix avait une incidence directe sur les résultats des différentes méthodes de segmentation, comme illustré sur les figures 2.10 et 2.11. Le *coefficient de mélange* est en effet exclu puisque sa mesure est indépendante de la segmentation de référence, seul le masque binaire de l’objet sert. Les tableaux 2.2 présentent les résultats obtenus sur la totalité des méthodes et des objets.

Ces chiffres confirment que la partition originale est fortement influencée par le choix du fond. En effet, le fond “blanc” a tendance à forcer la méthode à générer un sur-partitionnement artificiel qui ne se retrouvera pas une fois l’objet plongé dans des images de scènes réelles. Par contre, le choix du fond n’influencera pas les résultats de la mesure colorimétrique qui fait abstraction du sur-partionnement. Enfin, les fonds “chart” et “bruit”, qui sont issus d’une discrétisation similaire de l’espace des couleurs, sont plus corrélés entre eux que chacun pris indivi-

duellement avec le fond blanc.

2.4.2 Évolution en fonction des objets

Les figures 2.12, 2.13, 2.14 et 2.15 nous permettent de confronter les objets[2] d'une part dans l'espace mV, mHC et d'autre part dans l'espace mV, CM. Nous avons déjà notifié une absence de corrélation entre les différents descripteurs. Cependant, nous retrouvons bien un comportement fort différent suivant les méthodes. Avant de tirer des conclusions générales, analysons grossièrement les différents résultats individuellement selon chaque méthode:

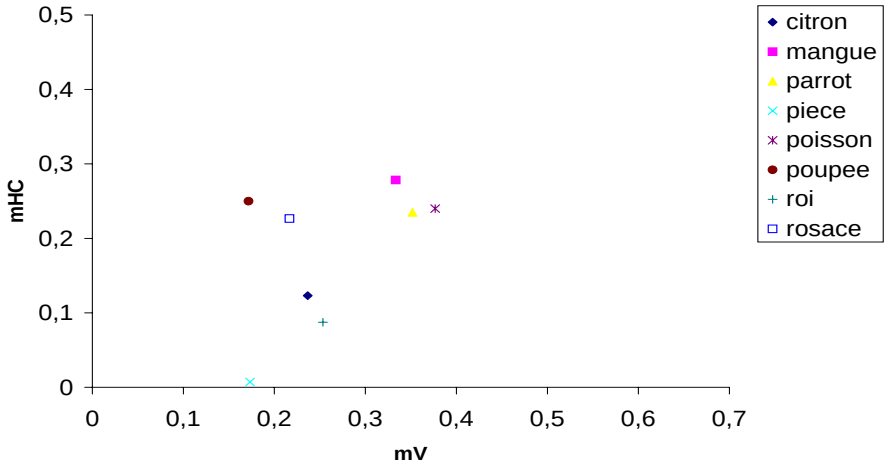
- MeanShift
  On note en premier lieu un coefficient de mélange élevé pour l'objet *pièce*. Le fait que ce dernier soit très peu coloré (quasiment monochromatique) et troué explique sans doute la "perte" régulière de cet objet. Par contre, sa mesure mHC est très bonne comme les autres objets mono-colorés. En revanche, sur un objet de type *mangue*, qui semble a priori un objet simple, la méthode n'obtient pas de bons résultats. On pourrait même rassembler les 3 objets *mangue, poisson et parrot* dans un petit groupe de mauvais élèves. Or, initialement, on ne peut pas considérer que la *mangue* soit aussi complexe que le *poisson*.
- WaterShed
  La première constatation est sans aucun doute les fortes valeurs pour la *mesure de Vinet*. Les mesures les plus faibles, pour *poisson* et *rosace*, sont quasiment supérieures aux plus fortes valeurs de la méthode meanshift par exemple. Autre remarque, le *citron* est l'objet le moins bien discriminé par cette mesure, alors qu'une seule région de référence la caractérise. La méthode watershed semble en effet incapable de le partitionner. Ainsi, le *citron* est recouvert par les mêmes régions sur moins de la moitié de sa surface. Pourtant, sa complexité semble toute relative. Dans une moindre mesure, les objets *pièce* et *citron* fournissent aussi des résultats peu convaincants.
- Clustering
  En faisant abstraction de la *mangue*, les objets ont tous plus ou moins la même valeur mHC mais des valeurs de mV et CM très différentes. Encore une fois dans une approche colorimétrique, c'est la *pièce* de par ses propriétes qui pose problème, dans le cas de la *mesure de Vinet*. Deux objets donnent lieu à des résultats plus convaincants: *rosace* et *poupée*. Les zones formant ces deux objets sont majoritairement monochromes et bien séparées les unes des autres. Ainsi la classification par nuées dynamiques est plus pertinente pour ces objets que pour des objets de texture plus marquée.

2. Pour le fond de type Chart

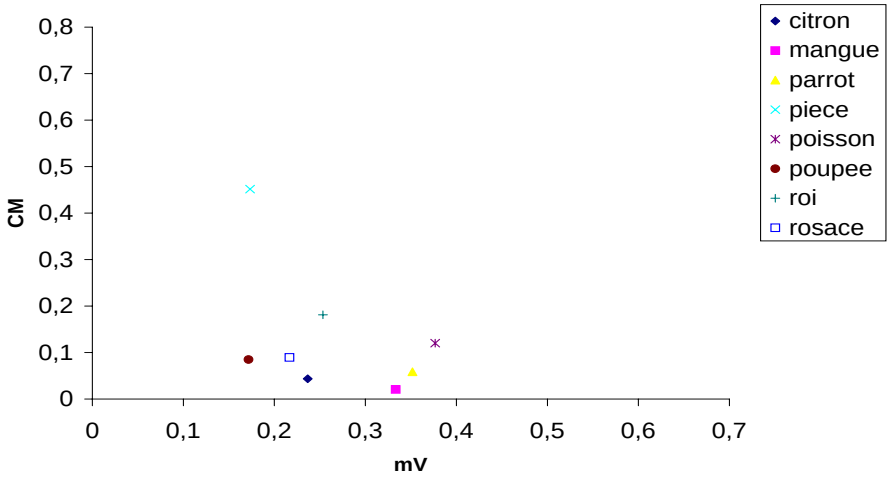
- Pyramide
  Cette méthode aussi, même si la mesure mHC est relativement stable, est très disparate quant aux autres mesures. Ce qui implique que si la partition spatiale évolue, l'objet est souvent mélangé à un fond de couleur très similaire. L'objet *citron* par exemple, de par sa forme en ellipse et son aspect mono coloré, est très bien adapté à une approche pyramidale. La méthode est la plus performante pour cet objet. En revanche, deux objets posent problème bien qu'ils soient mono colorés : le *roi*, de forme allongée, et la *pièce*, de forme ronde, mais trouée au milieu. La segmentation pyramidale, de par sa construction même, échoue souvent sur ce type d'objet[Bister *et al.*, 1990]. On peut a priori extraire 4 groupes d'objets au comportement similaire, {*citron*}, {*roi* et *pièce*}, {*poisson*} et {*mangue*, *parrot*, *rosace* et *poupée*}, qui ne semblent pas être pourtant liés. Plus globalement, la méthode semble donner des résultats moins satisfaisants que les autres sur les deux descripteurs morphométriques CM et mV.

Nous pouvions sans doute espérer, dans une optique d'abaque, que les méthodes basées sur l'analyse du nuage colorimétrique (clustering et meanshift) allaient conduire à un certain type de résultats, du moins autres que ceux issus des méthodes basées plus encore sur l'aspect spatial (pyramide et watershed). Même si certaines caractéristiques communes émergent, ce n'est pas toujours réellement le cas. Par exemple, pour les deux objets *mangue* et *poupée*, le coefficient de mélange varie très différemment selon que l'on se place dans une segmentation de type meanshift ou de type clustering, de même entre pyramide et watershed. Par contre, plus globalement, la mesure mHC est en moyenne plus faible pour les méthodes spatiales et la mesure mV plus faible pour les méthodes colorimétriques. Cela signifie que les méthodes spatiales, si elles ne segmentent pas toujours l'objet de la même façon, le font le plus souvent en respectant les couleurs de l'objet. Les méthodes colorimétriques se caractérisent également par un nombre de régions plus important. La segmentation semble alors spatialement plus stable, mais il arrive, comme précédemment expliqué, que certaines petites parties de l'objet contrastées soient, tantôt agrégées à une région plus grande, tantôt isolées, engendrant une mesure mHC relativement instable.

Généralement, une grande dynamique de la mesure de Vinet est notable, depuis des résultats satisfaisants ($mV = 0.10$) jusqu'à des résultats très médiocres (supérieurs à 0.5). Objectivement, cette valeur implique que seule une moitié de l'objet est correctement appariée avec la segmentation de référence. Dans ce cas, il sera alors très difficile de repérer un objet par une répartition spatiale, celle-ci évoluant trop largement. Globalement, il semble malgré tout correct de donner à meanshift une plus grande confiance pour son utilisation générique, l'ensemble des objets se positionnant à de plus faibles valeurs. Par contre, certaines méthodes sont plus efficaces dans certains cas précis (clustering pour l'objet *mangue* ou encore pyramide pour l'objet *citron*).

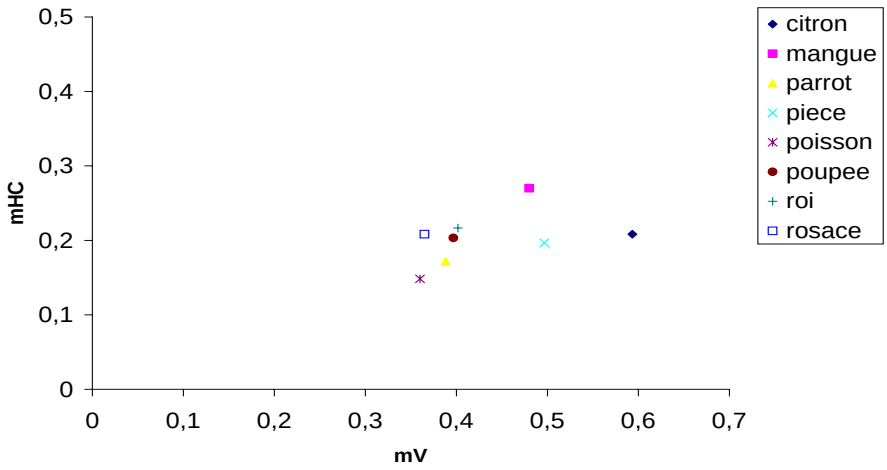


(a) mV, mHC

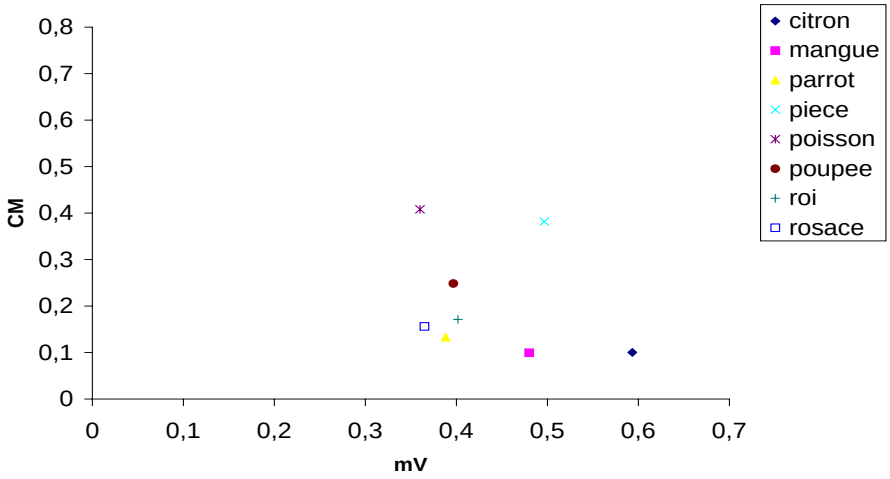


(b) mV, CM

Fig. 2.12 – *MeanShift : Positionnement des objets*

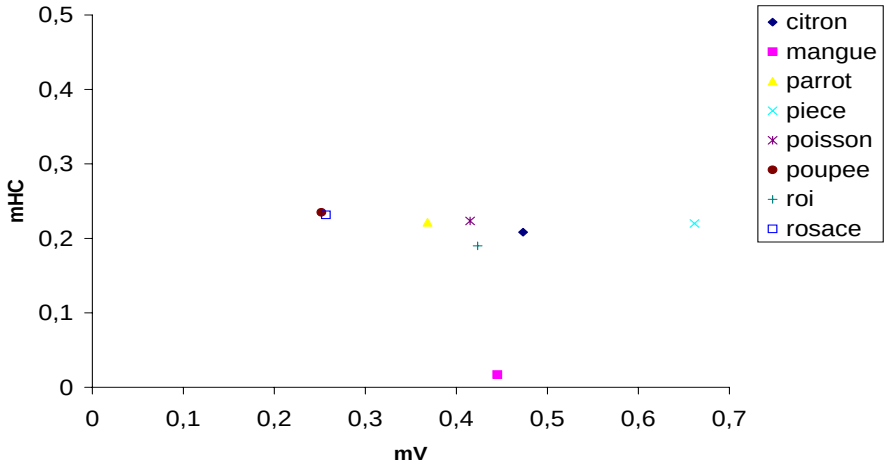


(a) mV, mHC

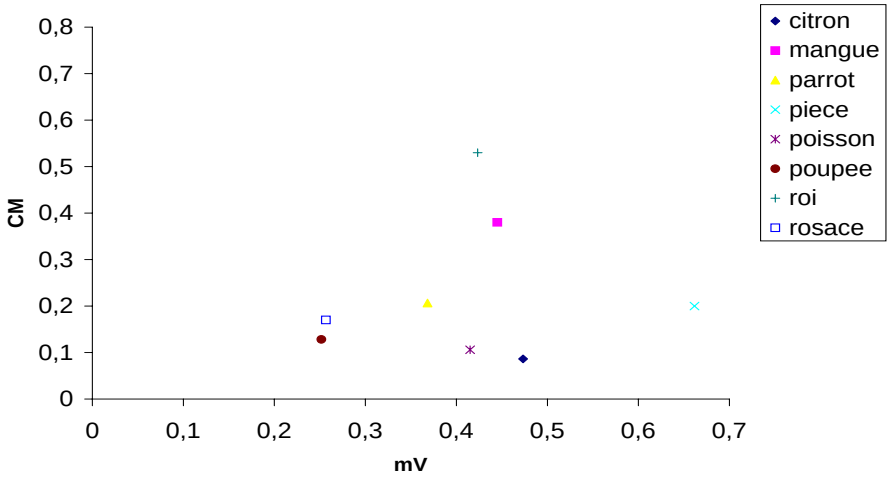


(b) mV, CM

Fig. 2.13 – *WaterShed : Positionnement des objets*

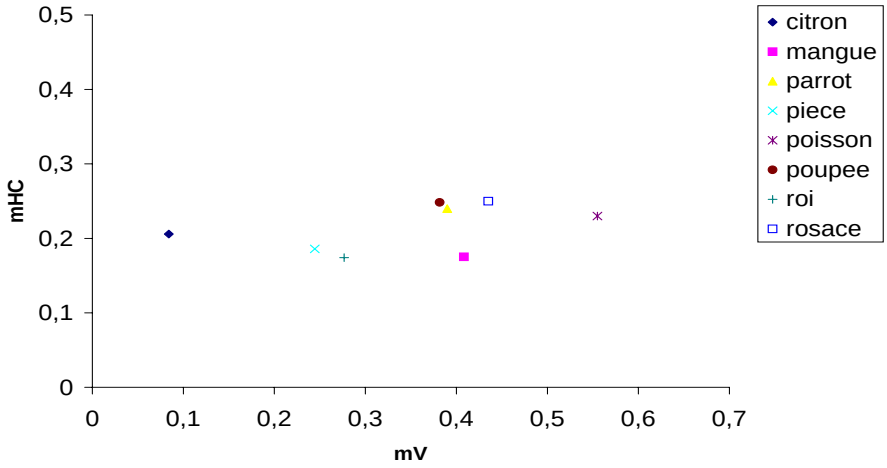


(a) mV, mHC

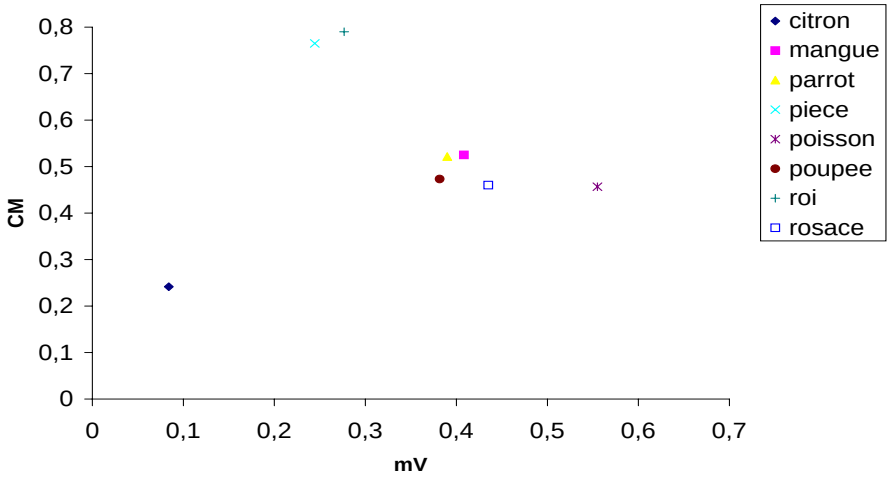


(b) mV, CM

Fig. 2.14 – *Clustering : Positionnement des objets*



(a) mV, mHC



(b) mV, CM

Fig. 2.15 – *Pyramide : Positionnement des objets*

Par contre, nous espérions pouvoir extraire des classes d'objets (mono/multi colorées et mono/multi texturées) plus ou moins bien traités par telle ou telle méthode. Or, il est de toute évidence difficile de retrouver sur ces graphiques une classification similaire, bien que certains regroupements apparaissent tout de même dans certains cas. En fait, la forme des objets semble aussi avoir une influence non négligeable, chaque méthode intégrant un côté plus-ou-moins spatial et pas uniquement un aspect colorimétrique. Deux objets forment néanmoins un groupe récurrent: *rosace* et *poupée*. Même si ces deux objets donnent des résultats qui varient sensible-

ment suivant la méthode considérée, ils sont toujours voisins sur les courbes précédentes. Le point commun entre ces deux objets est sans aucun doute de posséder des couleurs variées et des frontières fortement marquées, propriétés que l'on ne retrouve pas dans les autres objets.

2.5 Influence des fonds

Afin de mesurer plus précisément l'influence des différents fonds, les tableaux 2.3 illustrent les différentes valeurs obtenues pour chaque méthode suivant les fonds.

En premier lieu, il faut distinguer le fond blanc des deux autres fonds. Dans la majorité des cas, les deux fonds "chart" et "bruit" donnent des résultats similaires au contraire du fond "blanc" qui donne des résultats tantôt supérieurs tantôt inférieurs. Meanshift est sans doute la méthode la moins sensible au fond initial, les résultats étant souvent du même ordre. Néanmoins, certains écarts très importants sont notables sur la mesure couleur, comme dans le cas de l'objet *mangue* pour la méthode clustering ou l'objet *citron* pour la méthode watershed. Plutôt qu'une instabilité au contexte, c'est sans doute la segmentation de référence qui dans certains cas n'est pas du tout réaliste face aux plongements que l'on réalise.

Finalement, les tableaux présentés ici forment véritablement les résultats de notre protocole. En effet, afin de juger une méthode de segmentation, il faut, pour chaque fond et chaque objet type, connaître la mesure de Vinet et la mesure Histogramme-Couleur. Sans donner de recettes, calculer les valeurs mV et mHC sur des objets bien caractéristiques d'un problème de recherche, va permettre de choisir la méthode de segmentation a priori la plus adéquate. Les différent fonds sont primordiaux, car on ne peut se permettre de ne considérer qu'une seule segmentation de référence. À titre d'exemple, le fond *chart* est le plus adapté quant à la *rosace* pour la méthode watershed et le moins adapté quant aux autres méthodes. Même si ponctuellement, sur un objet ou une méthode, le choix du fond engendre un biais certain, les résultats globaux prouvent qu'en définitive le comportement de chaque méthode suit une seule et unique tendance. Une méthode moins performante ne deviendra pas plus performante globalement en changeant le fond de référence.

Maintenant, on désire évaluer le comportement des méthodes face à différentes variations que peut subir un objet. Nous allons donc, avant le plongement, modifier l'objet, sans toucher à sa forme géométrique, et évaluer de la même manière que précédemment les segmentations obtenues. Pour illustrer notre propos, nous avons choisi un fond de type *blanc* quant aux résultats affichés.

Objet	Fond	mV	mHC
Citron	Blanc	0.31	0.36
	Chart	0.41	0.0031
	Bruitée	0.26	0.13
Mangue	Blanc	0.44	0.26
	Chart	0.44	0.33
	Bruitée	0.31	0.28
Parrot	Blanc	0.36	0.23
	Chart	0.49	0.22
	Bruitée	0.25	0.23
Pièce	Blanc	0.16	0.0098
	Chart	0.16	0.003
	Bruitée	0.16	0.003
Poisson	Blanc	0.38	0.22
	Chart	0.40	0.23
	Bruitée	0.29	0.23
Poupée	Blanc	0.21	0.24
	Chart	0.21	0.25
	Bruitée	0.18	0.26
Roi	Blanc	0.28	0.13
	Chart	0.31	0.025
	Bruitée	0.35	0.17
Rosace	Blanc	0.30	0.22
	Chart	0.37	0.24
	Bruitée	0.21	0.22

Meanshift

Objet	Fond	mV	mHC
Citron	Blanc	0.15	0.14
	Chart	0.16	0.13
	Bruitée	0.48	0.25
Mangue	Blanc	0.26	0.28
	Chart	0.25	0.097
	Bruitée	0.45	0.011
Parrot	Blanc	0.49	0.26
	Chart	0.44	0.23
	Bruitée	0.35	0.23
Pièce	Blanc	0.61	0.15
	Chart	0.63	0.20
	Bruitée	0.65	0.23
Poisson	Blanc	0.38	0.23
	Chart	0.47	0.24
	Bruitée	0.40	0.23
Poupée	Blanc	0.30	0.25
	Chart	0.36	0.24
	Bruitée	0.23	0.24
Roi	Blanc	0.33	0.22
	Chart	0.30	0.13
	Bruitée	0.43	0.22
Rosace	Blanc	0.27	0.23
	Chart	0.30	0.24
	Bruitée	0.25	0.24

Clustering

Objet	Fond	mV	mHC
Citron	Blanc	0.39	0.072
	Chart	0.11	0.011
	Bruitée	0.30	0.18
Mangue	Blanc	0.22	0.30
	Chart	0.22	0.25
	Bruitée	0.41	0.27
Parrot	Blanc	0.43	0.17
	Chart	0.28	0.16
	Bruitée	0.25	0.17
Pièce	Blanc	0.60	0.15
	Chart	0.28	0.16
	Bruitée	0.36	0.20
Poisson	Blanc	0.52	0.14
	Chart	0.22	0.18
	Bruitée	0.39	0.16
Poupée	Blanc	0.48	0.18
	Chart	0.25	0.20
	Bruitée	0.18	0.22
Roi	Blanc	0.44	0.25
	Chart	0.44	0.25
	Bruitée	0.48	0.25
Rosace	Blanc	0.41	0.20
	Chart	0.12	0.15
	Bruitée	0.44	0.21

WaterShed

Objet	Fond	mV	mHC
Citron	Blanc	0.47	0.28
	Chart	0.59	0.29
	Bruitée	0.12	0.29
Mangue	Blanc	0.37	0.26
	Chart	0.43	0.25
	Bruitée	0.23	0.088
Parrot	Blanc	0.48	0.24
	Chart	0.39	0.27
	Bruitée	0.38	0.25
Pièce	Blanc	0.72	0.24
	Chart	0.23	0.27
	Bruitée	0.13	0.18
Poisson	Blanc	0.63	0.25
	Chart	0.50	0.25
	Bruitée	0.48	0.25
Poupée	Blanc	0.34	0.26
	Chart	0.44	0.28
	Bruitée	0.30	0.28
Roi	Blanc	0.25	0.22
	Chart	0.30	0.13
	Bruitée	0.24	0.055
Rosace	Blanc	0.50	0.26
	Chart	0.50	0.26
	Bruitée	0.36	0.26

Pyramide

Tab. 2.3 – *Influence des fonds sur l'ensemble des méthodes testées*

2.6 Évolution suivant différentes variations

Ce paragraphe présente différentes variations introduites pour évaluer la robustesse et la stabilité de la méthode de segmentation en fonction de l'objet. Ainsi cela va donner la possibilité de confronter les méthodes à des évolutions couramment introduites sur des images de scènes, à

savoir des:

- variations colorimétriques ;
- variations géométriques ;
- variations destructives.

De plus, étant donnée la complexité algorithmique relative au choix de 3 fonds distincts, du nombre d'objets et d'une quantité d'images suffisamment représentatives, nous limiterons cette étude par variation élémentaire, sans composer plusieurs variations entre elles. En effet, si l'on fait un calcul simple: nombre d'objets (8) $\times$ nombre de méthodes (4) $\times$ nombre d'images de scènes (3000) $\times$ nombre de types de variations (5) $\times$ nombre de variations pour chaque type (6) $\times$ temps de calcul moyen (10 secondes), on se rend compte que le temps de calcul est trop largement excessif (plusieurs années).

Voyons maintenant, variation par variation, les différentes évolutions que subissent les méthodes de segmentation. Les résultats sont présentés dans les sections 2.6.1,2.6.2,...,2.6.5 mais ne seront analysés que dans les sections suivantes, à savoir les sections 2.6.6 et 2.7. Néanmoins, nous ne nous lancerons pas dans une analyse point par point des résultats, analyse qui ne serait pas judicieuse et bien illusoire en l'absence d'un cadre applicatif précis. Le but n'est pas ici de donner une recette généraliste quant au choix des méthodes de segmentation pour la recherche d'objets mais de montrer que le protocole permet, dans le cadre d'applications plus scénarisées, de mieux appréhender les différentes méthodes envisageables.

2.6.1 Influence de la luminance

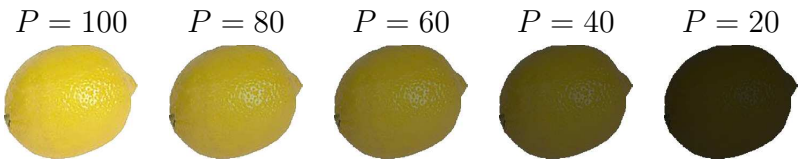


Fig. 2.16 – *Influence de la luminance sur l'objet "citron".*

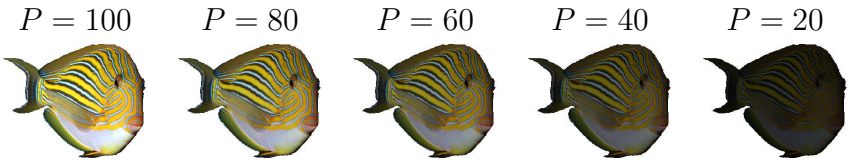


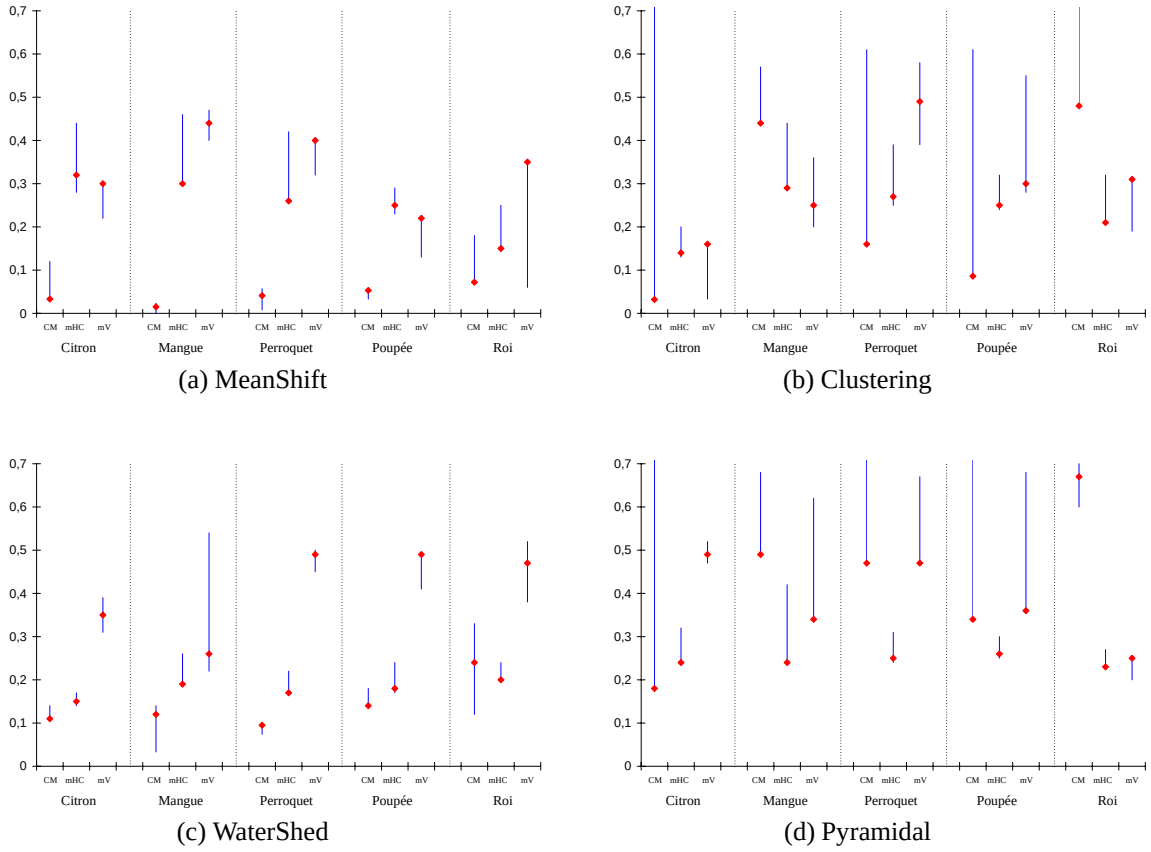
Fig. 2.17 – *Influence de la luminance sur l'objet "poisson".*

Chaque objet est plongé dans la base d'images après que sa luminance ait subie ,une variation. La luminance est ainsi multipliée par un pourcentage P, variant de 100% à 20% par pas de 20%; 100% correspondant à l'objet initial. Les figures 2.16 et 2.17 montrent l'influence de ce paramètre sur les objets citron et poisson.

Cette variation génère donc deux phénomènes :

- une compression de la dynamique couleur d'une part, d'où une plus grande difficulté à segmenter l'intérieur de l'objet. Les frontières intérieures sont en effet plus difficiles à discerner.
- une augmentation du contraste entre l'objet et son voisinage d'autre part, d'où une plus grande facilité d'extraction probable. Bien sûr, ce n'est vrai que dans le cas où l'image est globalement "éclairée". Si l'image est une image de nuit, alors diminuer la luminance ne permettra pas une meilleure extraction de l'objet. Néanmoins, dans notre base considérée, représentative d'une base généraliste, les images sont globalement des images prises de jour ou sous un éclairage intérieur. Du coup, pour la plus grande majorité, cette variation conduit à une extraction plus aisée de l'objet.

La figure 2.18 montre l'influence de la luminance sur les trois mesures CM, mHC et mV. Le point rouge correspond sur ces graphiques à la valeur initiale de la mesure. Le trait bleu représente quant à lui l'intervalle des valeurs prises par la mesure en fonction des différentes variations de luminance.



(a) MeanShift (b) Clustering

(c) WaterShed (d) Pyramidal

Fig. 2.18 – *Évolution en fonction de la luminance*

2.6.2 Influence de la teinte

Dans l'espace HSV standard, codé sur l'intervalle $[0,6[$, la composante H subit une variation de 0 à $+5$ par pas de 1. La figure 2.19 montre l'influence de ce paramètre sur l'objet *mangue*.

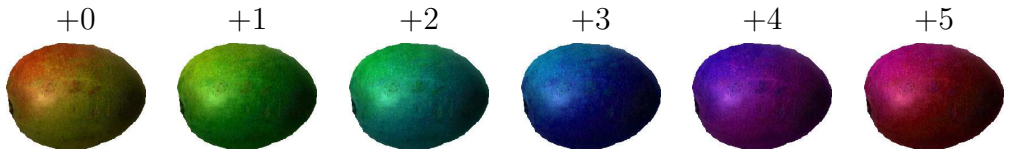
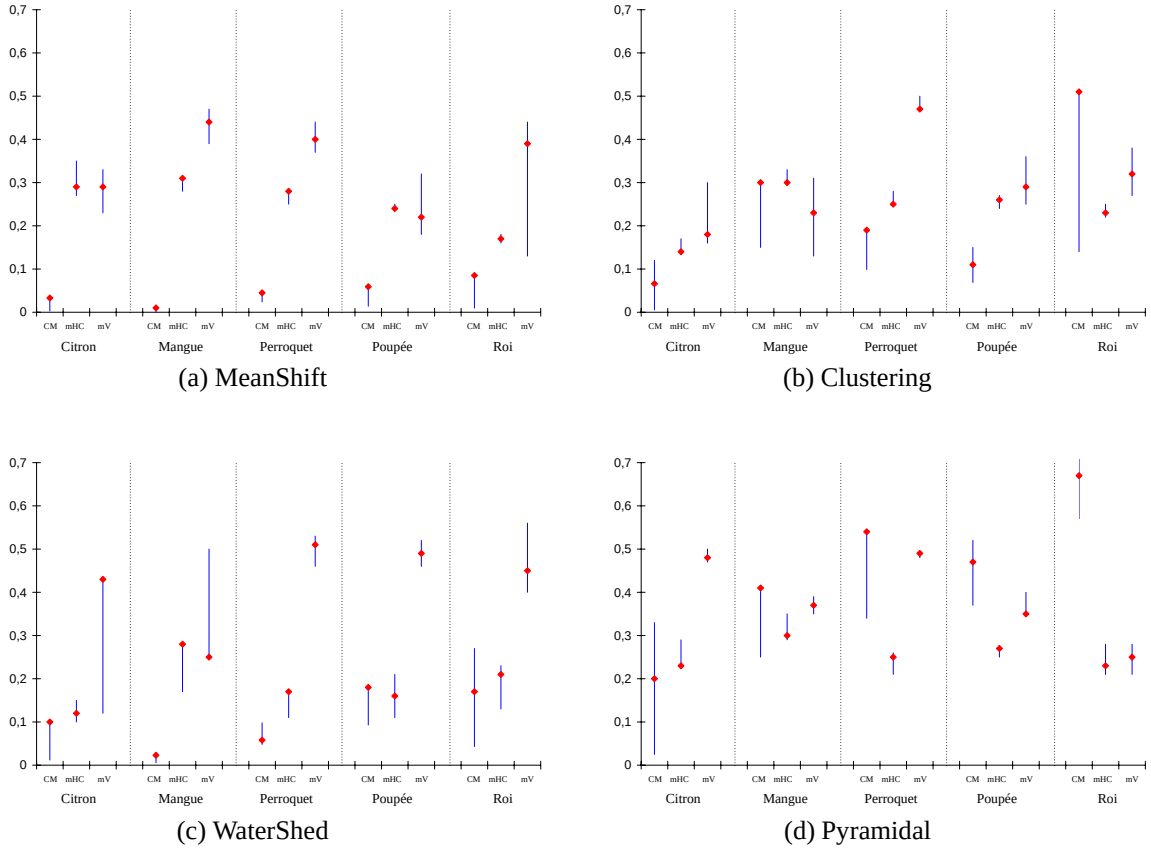


Fig. 2.19 – *Influence de la teinte sur l'objet "mangue".*

Ainsi le nuage colorimétrique est fortement perturbé bien que la quantité d'informations (l'entropie du nuage couleur) reste identique. Néanmoins, l'objet semble parfois perdre, visuellement en tout cas, pour certaines teintes une partie des contrastes internes qui le composent, comme pour l'objet *mangue*.

La figure 2.20 illustre l'influence de la teinte sur les trois mesures CM, mHC et mV.



(a) MeanShift (b) Clustering

(c) WaterShed (d) Pyramidal

Fig. 2.20 – *Évolution en fonction de la teinte*

2.6.3 Influence de l'illuminant

Une simulation de changement d'illuminant par adaptation chromatique est opérée sur l'objet [Finlayson et Süsstrunk, 2002]. Plus précisément, l'illuminant $D65$ constitue la référence et l'effet des illuminants $D50$, A, C, E et $F2$ est "simulé" sur l'objet. La figure 2.21 montre l'influence de ce paramètre sur l'objet *perroquet*.

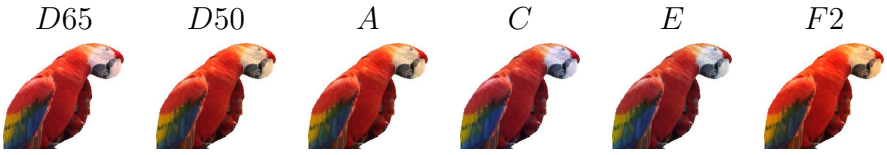
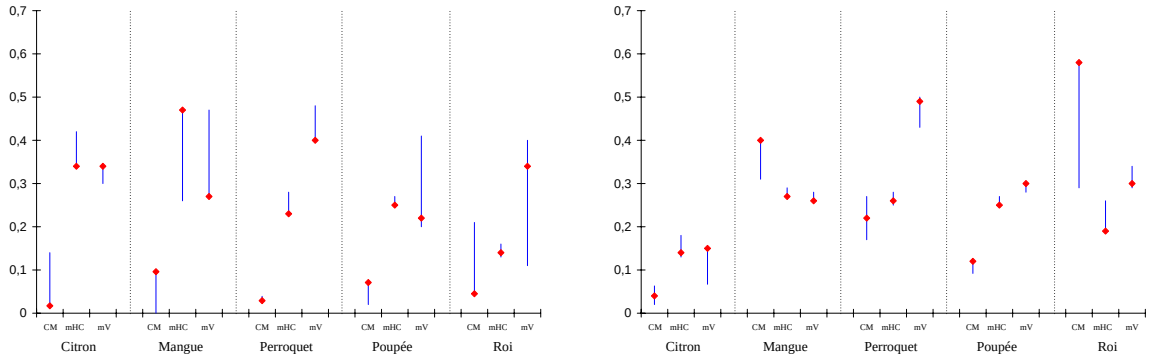


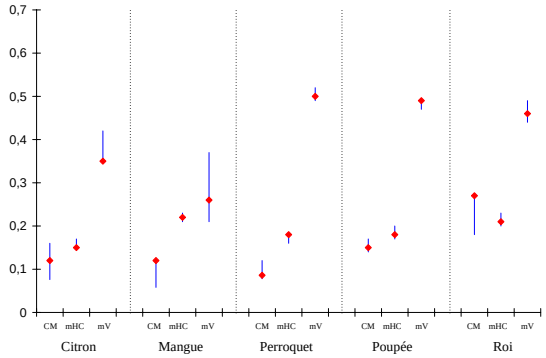
Fig. 2.21 – *Influence de l'illuminant sur l'objet "perroquet".*

A contrario du changement de teinte, le nuage colorimétrique est cette fois-ci légèrement perturbé. Surtout, la modification n'est pas une simple translation dans un espace couleur comme HSV mais bien un changement "réaliste" de l'apparence couleur de l'objet.

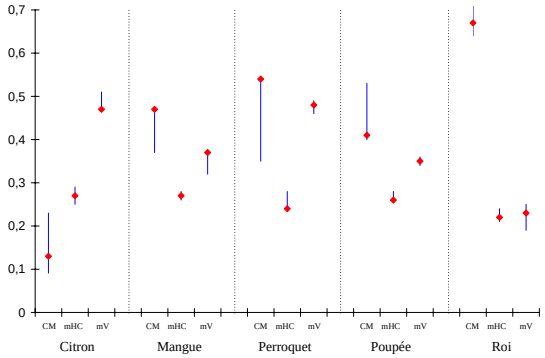


(a) MeanShift

(b) Clustering



(c) WaterShed



(d) Pyramidal

Fig. 2.22 – *Évolution en fonction de l'illuminant*

La figure 2.22 illustre l'influence de l'illuminant sur les trois mesures CM, mHC et mV.

2.6.4 Influence de la taille

Parallèlement, parmi les variations géométriques, nous avons uniquement retenu l'influence de la *taille* de l'objet, de 100 à 300 pixels dans la direction la plus grande par pas de 25. Une interpolation bi-cubique permet de calculer les différentes tailles sans créer d'artefacts qui nuiraient à la segmentation. Signalons que notre but n'est pas de rechercher un même objet quelle-que-soit sa taille ou son orientation [Torres-Mendez *et al.*, 2000] mais toujours de comparer un objet segmenté avec une référence extraite à la même échelle. L'évolution de la taille permet en fait de vérifier si la méthode va permettre de reconnaître un objet de façon identique quelle-que-soit la

proportion dans l'image occupée par l'objet. La figure 2.23 montre l'influence de ce paramètre sur l'objet *roi*.

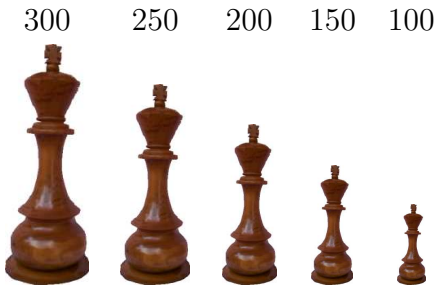
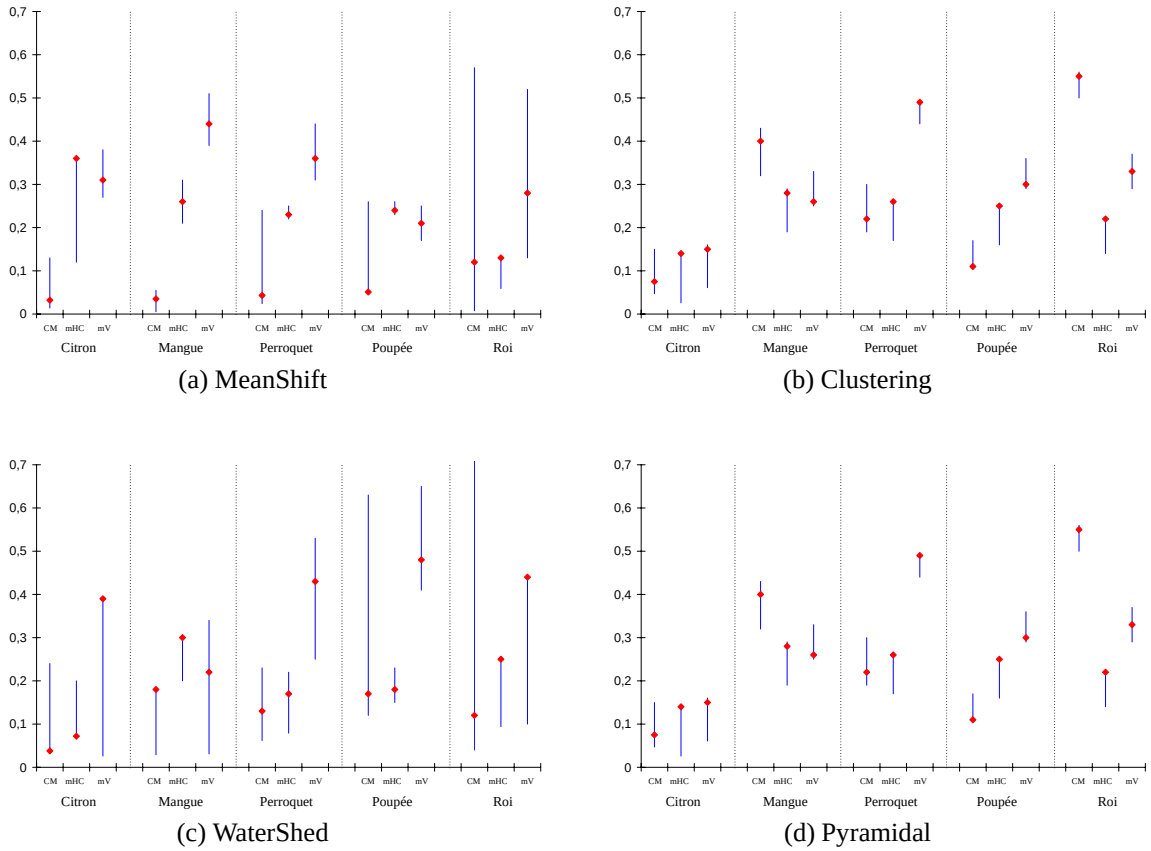


Fig. 2.23 – *Influence de la taille sur l'objet "roi".*



(a) MeanShift

(b) Clustering

(c) WaterShed

(d) Pyramidal

Fig. 2.24 – *Évolution en fonction de la taille*

La figure 2.24 illustre l'influence de la taille sur les trois mesures CM, mHC et mV.

2.6.5 Influence de la compression

Enfin, parmi les variations destructives, au sens qu'elles s'accompagnent d'une perte d'information, nous avons retenu l'influence de la compression JPEG parmi les autres approches possibles [Jolion et Bres, 1999], comme le filtrage, la quantification ou encore l'ajout de bruit. Des indices de qualité q de 100, 75, 50, 30, 15 et 10 sont utilisés. Pour ce paramètre particulier, étant donné le fait que cette compression engendre un effet de crénelage et donc de dégradation des frontières, l'image complète après plongement de l'objet est en fait compressée. La figure 2.25 montre l'influence de ce paramètre sur l'objet particulier *poisson* plongé dans une image de la base.

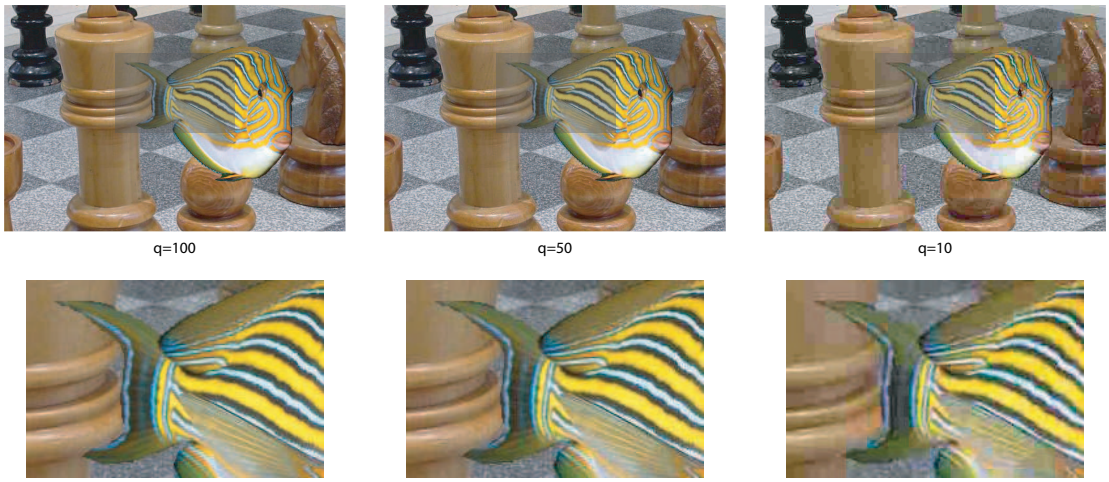


Fig. 2.25 – *Influence de la compression sur l'objet "poisson".*

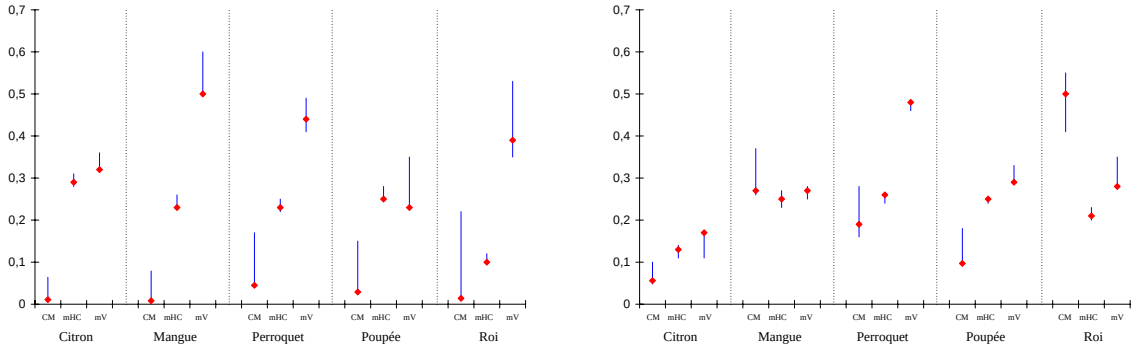
Sur cet exemple, il est à noter qu'une qualité faible donne néanmoins naissance à une image où l'information reste extractible a priori. Par contre, il est logique de voir que les frontières de l'objet perdent en contraste, la détection des régions va s'en trouver plus difficile.

La figure 2.26 illustre l'influence de la compression sur les trois mesures CM, mHC et mV.

2.6.6 Phénomènes classiques

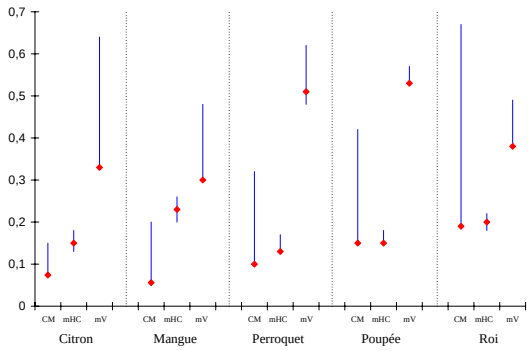
Tout d'abord, le protocole permet de retrouver certains résultats logiques et bien connus:

- La taille perturbe notablement le *coefficient de mélange*. En effet, plus l'objet est petit et plus l'outil de segmentation a tendance à le fondre au contexte. Cette propriété est plus marquée pour les méthodes *watershed* et *pyramide*, les deux outils pour lesquels le critère spatial est le plus accentué.

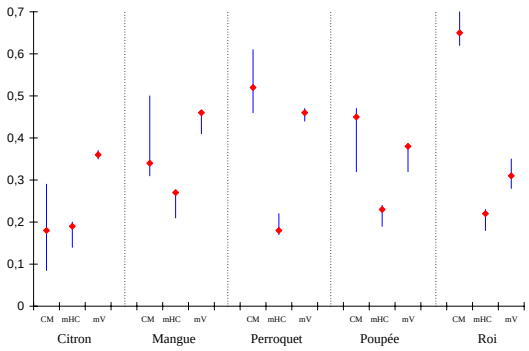


(a) MeanShift

(b) Clustering



(c) WaterShed



(d) Pyramidal

Fig. 2.26 – *Évolution en fonction de la Compression*

- La diminution de la luminance permet une meilleure extraction de l'objet, donc une diminution de la mesure *coefficient de mélange*. En fait, la valeur moyenne des images de scènes est assez stable et au centre de la plage de dynamique. Par contre, la segmentation interne de l'objet est quant à elle relativement perturbée, comme le démontrent les fluctuations de la *mesure de Vinet*.
- *watershed* est la méthode la plus sensible à la compression : les contours disparaissent lorsque le taux de compression JPEG augmente. *meanshift* et *clustering* sont plus stables : la compression JPEG respecte relativement bien la forme et la nature du nuage couleur.

2.7 Analyse des résultats

2.7.1 Une meilleure méthode ?

À l'issue de cette étude, il semble logique dans un premier temps de voir en *meanshift* la meilleure méthode. Ceci serait néanmoins faire abstraction de nombreux points. Tout d'abord, cette méthode donne le meilleur *coefficient de mélange*, et le meilleur résultat pour la *mesure de Vinet* dans la quasi majorité des cas, mais cette mesure reste influencée par l'ensemble des variations de façon non négligeable. En revanche, quant à la *mesure couleur*, moins influencée globalement dans le cas d'objets multi-colorés, *watershed* est souvent optimale, *clustering* et *pyramide* dans une moindre mesure donnant ponctuellement des résultats bien meilleurs. Ainsi, les descripteurs couleur calculés sur ces régions devraient être plus significatifs dans une optique de recherche de similarité.

2.7.2 Qualité des méthodes : suffisante ou insuffisante ?

Tout d'abord, il semble vain d'espérer construire un abaque pour contraindre le choix de la méthode de segmentation en fonction du type d'objets recherché. En effet, il n'est pas possible d'extraire dans une optique de recherche par le contenu, la meilleure adéquation entre l'objet et la méthode.

Ensuite, au regard de la relative faiblesse générale des méthodes testées dans le cadre de ce protocole ainsi que de leur instabilité aux différentes contraintes variationnelles, il est naturel de s'interroger sur la validité de cette approche segmentation/analyse dans le cadre d'une utilisation générique dans un contexte indexation d'images. En effet, résumons grossièrement les résultats :

- En moyenne 10% à 50% de l'objet est mélangé au fond environnant. Plus précisément, une mesure de *coefficient de mélange* de 25% signifie qu'en général un quart de l'objet n'est pas différencié de son contexte.
- En moyenne entre 20% et 40% du pavage spatial est perdu.

- L'information couleur n'est que partiellement restituée mais elle l'est de manière stable.
- Les variations non destructives ont une influence non négligeable.

L'utilisation d'une approche segmentation/reconnaissance de régions n'est ainsi pas encouragée à la vue de ces conclusions. Au contraire, un certain pessimisme serait plutôt de rigueur, tant les outils de segmentation testés ne présentent pas a priori une stabilité suffisante pour de telles approches. A contrario, rappelons que nous avons réglé les méthodes de segmentation une fois pour toutes, dans notre optique benchmarking, en ne retenant toujours que les réglages qui offraient les meilleurs résultats globaux, mais nous n'avons aucunement cherché à affiner chacune d'entre elles image par image.

Aboutissant à la conclusion classique qu'aucune méthode de segmentation ne semble toujours être la plus appropriée quel que soit le contexte envisagé, nous avons malgré tout cherché au cours de cette étude à évaluer de façon objective cette étape pour une utilisation générique en indexation d'images. Notamment en introduisant un protocole de validation au cours duquel des objets quelconques subissent des contraintes extérieures, depuis des variations colorimétriques, géométriques jusqu'à des variations destructives. Les résultats présentés peuvent néanmoins conduire à abandonner certaines méthodes dans le cas de certains scénarii. De plus, les méthodes testées, tantôt colorimétriques tantôt spatiales, ne permettent pas de conduire à une sélection probante de l'approche en fonction d'une répartition spatio-colorimétrique représentative de l'objet. En effet, les familles d'objets ne se regroupent pas dans des classes bien discriminées au niveau des descripteurs retenus. Ceux-ci traduisent en fait la capacité d'une méthode à bien discriminer l'objet de son fond environnant, à bien respecter le pavage dans différents contextes et à respecter le nuage colorimétrique de l'objet. Le tableau 2.4 résume rapidement les avantages et inconvénients de chacune des différentes méthodes qui ont illustré ce protocole.

Malgré tout, la méthode qui semble globalement la plus efficace et la plus stable aux différentes variations, ie l'algorithme du meanshift, n'impose pas sa supériorité de manière unanime. Dans certains cas précis, rien ne permet de conclure qu'une autre méthode ne serait pas plus opérante. Néanmoins il devient en définitive possible, en connaissant les valeurs des mesures CM, mV et mHC, d'estimer "grossièrement" la qualité des résultats que l'on peut attendre de la phase de reconnaissance, et ainsi de donner plus d'influence à certains descripteurs calculés sur chaque région ensuite. Si, par exemple, la mesure de Vinet est très instable, il ne faudra pas donner trop de confiance à des descripteurs de type forme.

2.8 Perspectives

Nous avons essayé de proposer un protocole d'évaluation objectif de l'outil habituellement situé en amont de la phase de recherche elle-même, à savoir la segmentation. En effet, cette

Méthode	Remarques
Meanshift	• Globalement la plus effi cace • Grand nombre de régions • Relativement coûteuse en temps • Bonne stabilité de la partition spatiale • Coeffi cient de mélange bas mais mesure Couleur parfois élevée • La moins sensible aux diverses variations et aux fonds
Clustering	• Faible coût en temps • Résultats parfois non satisfaisants • Peu sensible à la compression • Peu effi cace pour les objets mono colorés • Peu sensible à l'illuminant • Peu sensible à la taille • Grande sensibilité à la luminance
Watershed	• Relativement coûteuse en temps • Mesure Couleur souvent faible et stable aux variations • Tendance à perdre les petits objets • Partition spatiale souvent non respectée et très sensible aux variations • Peu adaptée aux objets avec des dérives colorimétriques
Pyramide	• Faible coût en temps • Non satisfaisante sur les objets multi-texturés • Très sensible à la forme des objets • Très sensible aux changements de luminance • La plus stable à la compression • Généralement la moins probante des méthodes

Tab. 2.4 – *Récapitulatif des méthodes testées*

méthode est antérieure à toute recherche de similarités. Nous introduisons ainsi des mesures objectives permettant de juger de la stabilité d'une méthode de segmentation et de sa capacité future à être exploitée dans un contexte de calcul de similarités. Certaines méthodes couvrant au maximum la gamme des techniques en usage (clustering, watershed, meanshift, pyramide) ont été testées et critiquées selon ce protocole d'évaluation. Si chaque méthode semble trouver un cadre applicatif pour lequel elle est la plus efficace, la méthode meanshift semble néanmoins la meilleure et la plus stable à certaines contraintes qui pourraient en pratique nuire à l'extraction d'information sur les images. Malgré tout, la relative faiblesse des résultats semble conduire à la nécessité d'établir un lien actif entre certaines méta-données et la phase de segmentation bas-niveau afin de contraindre cette dernière.

De plus, nous pensions qu'il serait possible d'établir des classes d'images pour lesquelles telle ou telle méthode de segmentation serait la plus appropriée. Nous avons donc cherché à construire des paramètres suffisamment simples et peu coûteux en temps, plutôt d'ordre colorimétrique que spatial, pour contraindre le choix de la méthode à employer. L'exploitation des résultats actuels ne nous a pas permis pour l'instant de répondre favorablement à cette requête mais cette orientation reste dans la perspective de nos travaux actuels.

Troisième partie

Applications

Chapitre 1

DÉTECTION DU FLOU DANS LES IMAGES DE SCÈNES

Sommaire

Extraire des méta-données sur une image est sans aucun doute l'une des voies à accentuer afin de capitaliser de l'information sur une image avant même de rechercher une similarité. Par exemple, reconnaître dans une image où se trouve le focus d'attention semble approprié. Dans ce contexte, nous proposons un détecteur de zones floues d'une image. Suite à une segmentation en régions grossières, un algorithme d'apprentissage permet, à partir de descripteurs classiques, de conduire à une séparation en zones floues et nettes.

1.1 Avant-propos

Le système visuel identifie une image à partir des éléments qui la composent. Le réseau cortical sous-jacent à ces processus d'identification utilise entre autres des informations dites de bas niveau, telle que la forme, la couleur, la texture… des élements composants l'image pour initialiser l'identification [Mel, 1997]. Des éléments annexes à l'image vont compléter ensuite la construction sémantique de l'image. Cependant, ces éléments peuvent soit apporter des indices visuels qui peuvent optimiser la reconnaissance de l'image par une fusion cohérente de ces deux types informations, soit contraindre l'identification car les informations ne sont pas concordantes.

Ces éléments annexes peuvent, par exemple, être définis par la présence de perspectives, d'ombres, de volume mais aussi par la présence de "flou". Le flou est un indice visuel qui exprime entre autre une notion de profondeur qui permettra par exemple de délimiter rapidement les divers plans de l'image [Mather et Smith, 2000]. Cette segmentation des éléments participe positivement à la reconnaissance de l'image et ce, grâce à l'extraction de la zone informative de l'image.

Fig. 1.1 – *Une classique photographie de vacances*

Ainsi, dans un objectif d'indexation d'images par le contenu pixelique, il semble naturel de distinguer les pixels des régions floues des autres. L'information sémantique de la partie floue de l'image ne peut pas être considérée de la même manière que celle de la partie nette. Si, par exemple, un père de famille prend en photographie sa fille sur un fond montagneux (figure1.1), les deux informations "sa fille" et "fond montagneux" sont portées distinctement par les régions nettes (focus de l'appareil photographique sur la fille) et les régions floues pour le décor montagneux. La détection du flou est donc très importante pour distinguer :

- L'objet ou la personne qui est photographié, sur qui le focus se porte;
- Le contexte ou une partie du contexte qui entoure cet objet.

Cependant, il est prudent de disinguer plusieurs types de flous. Selon l'origine du flou, de son importance dans l'image, de sa localisation au sein des éléments de l'image, la participation sémantique du flou peut être différente lors de la reconnaissance de l'image. Nous pouvons tout

(a)

(b)

(c)

Fig. 1.2 – *Différents types de flou*

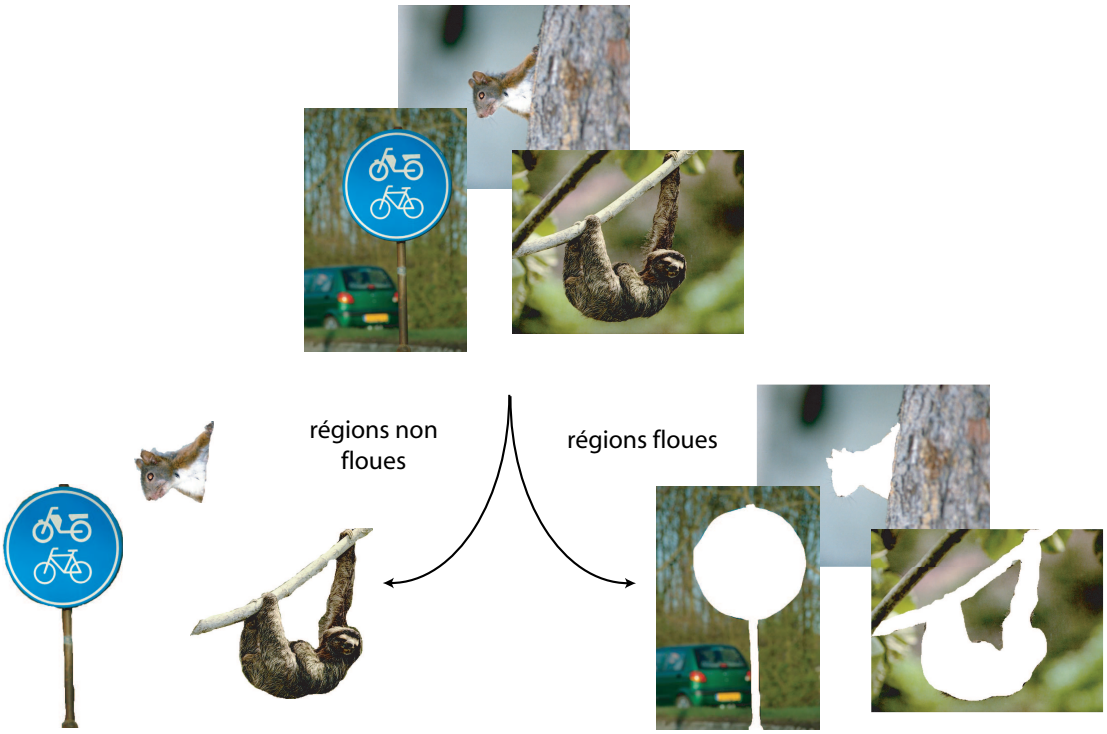


Fig. 1.3 – *Séparation flou non flou*

d'abord distinguer, en partant du général au local, le flou "global" qui dégrade toute la scène visuelle (induit par un défaut de vision par exemple) : C'est un flou homogène qui freine la reconnaissance de l'image car le système visuel ne peut pas distinguer les élements de l'image. Nous pouvons ensuite distinguer les flous présents seulement sur une ou plusieurs zones de l'image. Ce sont des flous qui peuvent être induits par le mouvement des objets comme pour la voiture de course sur l'image1.2a. La physiologie du système visuel ne permet pas toujours de supprimer instantanément cet effet. Une deuxième source de type de flou est la défocalisation dioptrique (exemple de l'appareil photo). Ces deux dernières catégories de flou permettent de guider la reconnaissance de l'image contrairement à la première. La différence entre les deux est liée à la sémantique apportée du flou. Dans le premier cas, le flou couvre et encombre le contenu de l'image. Dans le deuxième cas, le flou definit notamment un repère spatial des différents éléments constituant l'image (exemple du flou induit par le mouvement), ou un renforcement du contour ou des bords de l'objet sélectionné par le focus (cas de l'appareil photo).

Ainsi, dans notre étude[1], nous nous intéresserons uniquement à l'analyse de ces deux dernières catégories de flou: l'objectif étant d'extraire, comme le montre la figure 1.3, les zones d'intérêt grâce à la détection du flou qui les entoure. Ainsi une sémantique induite pourra enrichir la recherche de similarité postérieure.

1. Ce travail a fait l'objet d'une publication: [Da Rugna et Konik, 2004a]

Parallèlement, la détection du flou dans les images, ou bien dans une vidéo, peut se révéler opportune dans de nombreux autres cas. L'extraction de méta-données, comme l'extraction de texte sur les images[Wolf et Jolion, 2003], au niveau de la norme MPEG-7 pour les vidéos par exemple, serait sans aucun doute un apport non négligeable pour les diverses traitements liés à l'affichage ou au re-cadrage. Une autre application de la détection du flou intervient dans le cadre de la vision humaine et plus précisément dans le cadre de la réhabilitation visuelle de personnes souffrant d'une altération de la vision centrale (souvent liée à une dégénérescence maculaire liée à l'âge). Cette maladie engendre une grande difficulté dans l'identification d'images: la perception de celles-ci est partiellement, voir totalement, floue. Le malade ne peut ainsi pas extraire l'indice "flou" et distinguer du coup le flou du net de l'image. La reconstruction de la scène visuelle par le cerveau sera plus difficile. La détection des parties floues permettrait alors de proposer une présentation différente de l'image. Par exemple, il sera possible de guider la reconnaissance de ces patients en renforçant les objets d'intérêt de la scène.

Pour extraire les parties floues d'une image et, a fortiori les parties nettes, on peut envisager diverses stratégies. Il semble évident qu'une approche locale (ou pseudo locale) soit nécessaire : une décision du type "cette image contient du flou", sans spécifier quelles zones sont floues, n'est pas suffisante. On peut alors définir deux types d'approches :

- Pixels
  On cherche à définir, pour chaque pixel un degré de flou, en examinant sa valeur par rapport à son entourage. Ensuite, un ensemble de pixels adjacents "suffisamment" flous sera alors considéré comme région floue. De l'information pixel, on déduit l'information région.
- Régions
  Une étape de segmentation permet d'extraire des régions bien distinctes. Chaque région est alors caractérisée par différents descripteurs qui permettront de définir la notion de flou ou de net. Le flou est ainsi extrait en considérant la région dans son ensemble.

Notre choix s'est porté sur l'approche région pour diverses raisons. Dans notre optique, une région floue n'est pas une concaténation de pixels considérés comme flous. C'est la région prise dans son ensemble qui est considéré comme floue. Plus précisément, un pixel ne peut être flou que s'il appartient à une région floue. De plus, une approche région semble plus performante en terme de complexité : l'approche pixel oblige tout d'abord à quantifer le flou pour chaque pixel et ensuite à étendre les régions pixels à pixels.

Néanmoins cette approche ne peut être perspicace que pour certaines conditions quant à la segmentation. Avant de revenir sur ce point, listons brièvement les principaux points de notre démarche illustrée par la figure1.4.

- Étape de segmentation
  Une segmentation efficace est primordiale pour réussir ensuite une bonne détection. L'étape

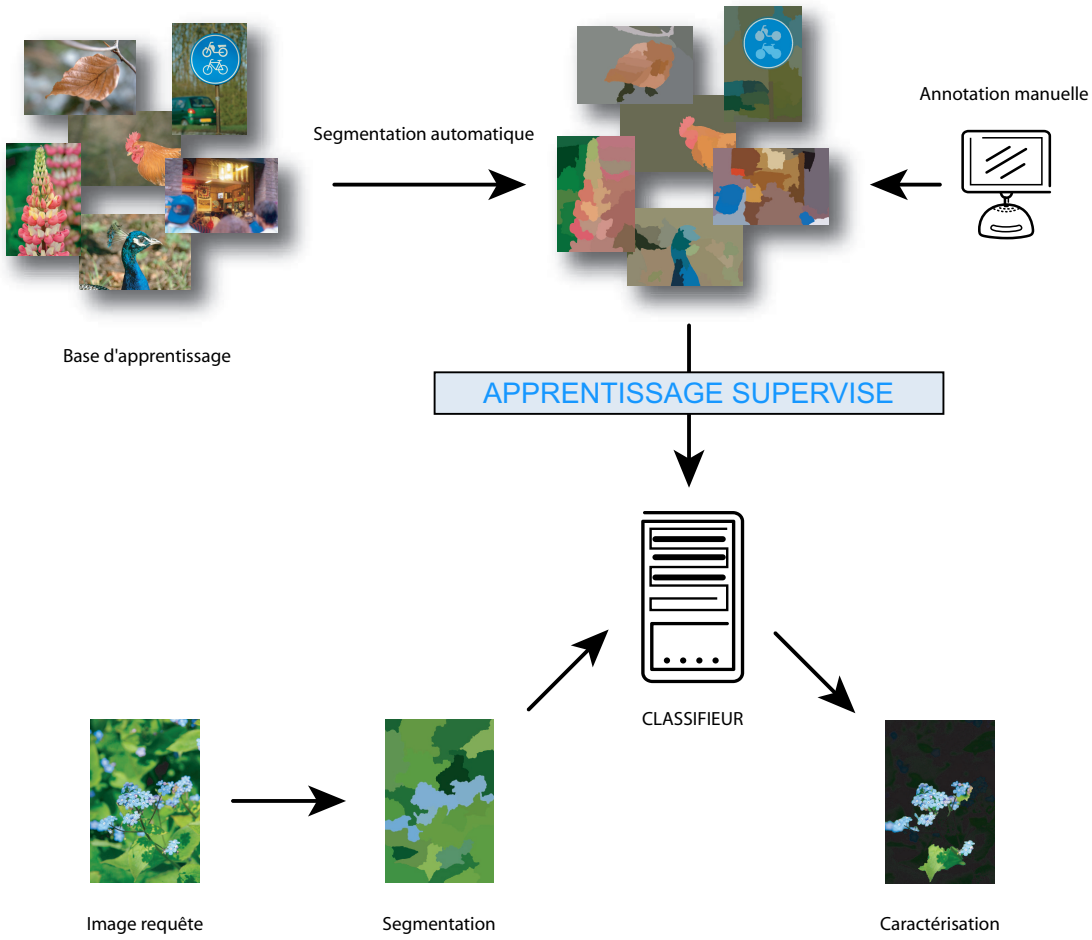


Fig. 1.4 – *Principe général de la détection du flou*

de segmentation doit fournir des régions :

- homogènes en couleur et en texture pour ne pas perturber les descripteurs.
- ne recouvrant qu'un seul type de flou : flou optique, mouvement, net... Si une zone floue et une zone nette sont regroupées dans une même région la détection n'est pas possible.
- relativement grandes. Un morcellement excessif entraînerait, en plus d'une plus grande complexité, le risque de biaiser les résultats sur les petites régions, peu riches en informations.

- Description des régions
  Chaque région doit être décrite par une série de descripteurs. Ces derniers doivent être corrélés avec l'information flou. Ils sont obtenus notamment par le calcul des différents moments de descripteurs pixeliques : moyenne, écart-type et autres paramètres de texture...
- Classification
  En fonction des caractéristiques de chaque région, un classifieur doit permettre de qualifier les régions comme floues ou non, avec un taux de confiance quant à la qualité de la classification.

1.2 Étape de segmentation

Après avoir choisi de passer par une étape bas niveau de segmentation, se pose alors la question de l'outil proprement dit. Bien évidemment le large panel d'algorithmes, évoqué en partie II, menant à une segmentation de l'image, nous oblige à une pré-sélection. Les exigences quant à la segmentation excluent de facto les méthodes basées sur le nuage colorimétrique. En effet, les segmentations basées sur une analyse du nuage couleur, qu'elles intègrent ou non une légère connotation spatiale, ne sont pas adaptées à notre problématique: le risque d'agrégation de zones floues et nettes dans une même région est trop important et ne permettrait pas une bonne détection. En effet, par exemple dans le cas d'une segmentation par nuées dynamiques, une fois les germes sélectionnés, il ya une forte probabilité qu'une zone floue soit agrégée avec une zone frontalière nette de même germe couleur : il y a donc échec de la segmentation pour une détection du flou. On retrouve ce même travers dans le cas d'algorithmes comme le Mean-Shift, ou bien encore ceux basés sur les histogrammes.

Grossièrement on peut considérer que les régions floues sont moins influencées par les filtres passe bas que les autres régions. Ce qui amène dans un premier temps à introduire une structure pyramidale pour réaliser cette segmentation. En effet, l'idée principale d'une Pyramide gaussienne est de produire une série d'images à des résolutions progressivement réduites. Chaque

niveau est obtenu via un filtre passe-bas et en même temps un sous-échantillonnage ce qui semblerait bien adapté à la détection de régions floues.

Parallèlement à ceci, les frontières entre les régions floues et non floues, même si elles ne semblent pas très marquées, sont quand même existantes visuellement. Une approche morphologique semble convenir à une détection de ces frontières : l'approche ligne de partage des eaux[Vincent et Soille, 1991, Andrade *et al.*, 1997].

La figure 1.5 présente des résultats de ces deux méthodes sur diverses images contenant du flou. Généralement la méthode pyramidale a tendance à fournir de plus nombreuses régions. Néanmoins, les deux méthodes souffrent parfois du mélange dans une même région d'une partie floue et d'une partie non floue. Bien-sûr, cette confusion va perturber les étapes suivantes d'apprentissage: les paramètres issus de ces régions mixtes seront forcément des éléments irréguliers par rapport à la phase d'apprentissage automatique. Notons par contre que dans le cas de mixité de flou, on attend à ce que le système réponde "non flou". On ne désire en effet détecter (différencier dans une future étape d'indexation ou d'archivage) que les régions floues très majoritairement voir totalement. Une région mixte contient de l'information de type nette et de l'information de type floue. Néanmoins, nous avons choisi de privilégier l'information centrale, celle des zones nettes.

1.3 Les descripteurs

En considérant que l'étape de segmentation est suffisamment robuste, chaque région doit être quantifiée par une série de descripteurs adaptés à la reconnaissance du flou. Chaque descripteur doit apporter son pouvoir discriminant pour permettre à une méthode d'apprentissage de les faire coopérer. Dans un premier temps, nous allons donc introduire un grand nombre de paramètres issus de diverses approches. De ce large panel de descripteurs proposés, il sera ensuite possible de sélectionner les plus discriminants.

L'idée de base, certes naïve, de cette étude est que les régions nettes sont plus sensibles aux effets des filtres passe-bas. La figure 1.6 illustre bien le phénomène, où la région non floue proposée en exemple est bien plus influencée par le filtre passe-bas que la région déjà floue. En revanche, il semble difficile de quantifier directement cette sensibilité. Ainsi une évolution relative, avant et après un traitement, semble être plus justifiée dans notre démarche.

Ainsi, différents paramètres sont associés à chaque région issue de la segmentation et ceux-ci seront quantifiés via leur évolution par rapport à un filtre passe-bas. Trois grandes familles de descripteurs ont été retenues :

- Évolution des moments statistiques
- Évolution de longueurs de plages

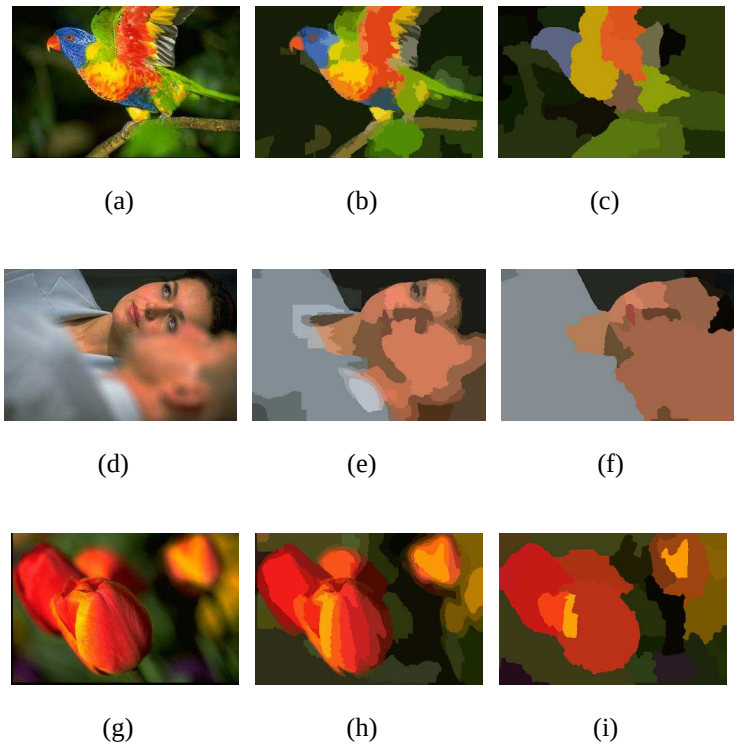


Fig. 1.5 – *Segmentations obtenues avec la méthode Pyramide et la méthode WaterShed - image originale à gauche.*

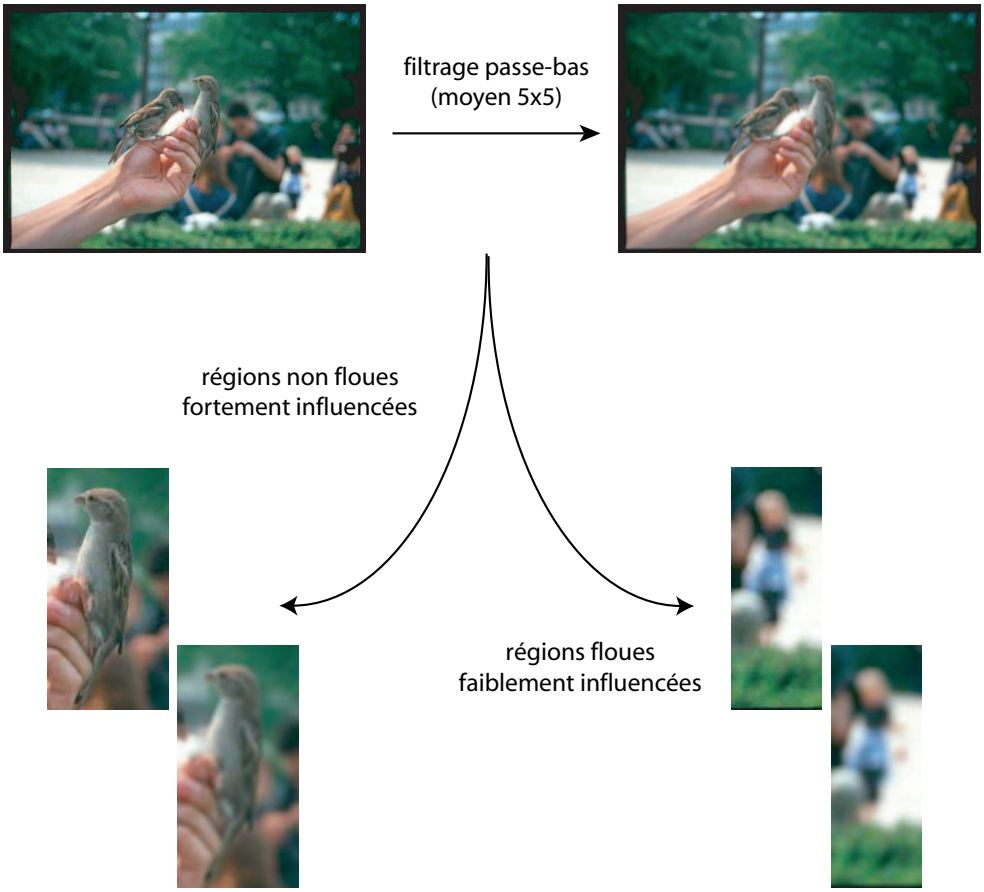


Fig. 1.6 – *Effet du filtrage passe-bas sur les régions floues*

- Évolution des hautes fréquences

1.3.1 Moments statistiques

Les quatre premiers moments sont calculés pour chaque région de l'image initiale. Sur différentes images, résultant d'un filtre passe-bas (de 3×3 à 9×9), les mêmes moments ont été calculés, sur les mêmes régions géométriques.

On note $\mathcal{M}_n^o$ et $\mathcal{M}_n^F$ les moments d'ordres n de la région d'origine et de la région correspondante à l'image filtrée.

Pour chaque filtre on calcule l'évolution relative :

$$\frac{\mathcal{M}_n^F - \mathcal{M}_n^o}{\mathcal{M}_n^o} \tag{1.1}$$

Ainsi chaque région est représentée par la valeur de cette évolution relative. Notons ici qu'une observation des résultats, pour les moments ou pour les autres descripteurs, montre qu'il n'est

pas possible de spécifier un seuil qui distinguerait le flou du net. On ne peut malheureusement pas expliciter une règle simple du type : Si $Moment\ d'ordre\ 2 > Seuil$ alors la région est floue.

1.3.2 Descripteurs de texture

Les nombreuses études existantes dans la caractérisation de texture, des ondelettes aux matrices de co-occurrences, nous ont donné un large choix pour le descripteur de textures à associer à notre approche. Nous avons opté pour les matrices de longueurs de plages qui mesurent les irrégularités locales autour de chaque pixels. Plus précisément, ici, nous décrirons l'adaptation couleur[Vertan *et al.*, 2002a] que nous avons évoquée en partie I. Les paramètres issus de ces matrices semblent ainsi bien adaptés à la détection de zones "floues" car leurs régularités, ou leurs irrégularités doivent permettre de les différencier. Afin de réaliser le calcul proprement dit des longueurs de plages une quantification préalable est nécessaire. Pour cela nous avons sélectionné deux quantifications couleurs avec des caractéristiques distinctes :

- Quantification de type Mean-Shift[Comaniciu et Meer, 2001]: quantification respectant au mieux les caractéristiques visuelles de l'image en adaptant le nombre de couleur en fonction de l'image.
- Quantification de type nuées dynamiques: quantification en un nombre pré-défini de couleurs.

Rappelons à présent brièvement la définition des matrices de longueur de plages : construction, pour une direction donnée, d'une matrice non pas de co-occurrence mais de continuité d'une couleur donnée.

Soit C le nombre de couleurs de l'image et T la longueur maximale dans la direction Θ. La matrice est de taille $C \times T$. Notons que pour chaque région, une matrice de longueur de plages est calculée. Soit $L^{\Theta}[i,j]$ le nombre de sections de couleur i et de longueur j. On entend par section une suite de pixels de même couleur i dans la direction Θ : j pixels de couleur i sont donc adjacents dans la direction Θ.

Différents paramètres sont extraits de cette matrice de longueur de plages. Dans le cadre de notre étude, trois se sont avérés les plus discriminants :

- Poids des petites sections: $\dfrac{1}{\sum_{i=1}^{C}\sum_{j=1}^{T} L^{\Theta}[i,j]} \sum_{i=1}^{C}\sum_{j=1}^{T} \dfrac{L^{\Theta}[i,j]}{j^2}$
- Poids des longues sections: $\dfrac{1}{\sum_{i=1}^{C}\sum_{j=1}^{T} L^{\Theta}[i,j]} \sum_{i=1}^{C}\sum_{j=1}^{T} L^{\Theta}[i,j] \times j^2$

- Homogénéité : $\frac{1}{\sum_{i=1}^{C}\sum_{j=1}^{T} L^{\Theta}[i,j]} \sum_{i=1}^{C}\left(\sum_{j=1}^{T} L^{\Theta}[i,j]\right)^2$

Bien entendu, les directions privilégiées pour des images de scène sont 0° et 90° . Cependant, il nous a semblé intéressant d'utiliser un plus grand nombre de directions: le calcul se fait donc de 0° à 315° par pas de 45° .

En définitive, chacun des trois paramètres est donc calculé pour $\Theta = 0^\circ$ et $\Theta = 90^\circ$. De plus les valeurs minimum et maximum sur l'ensemble des angles sont aussi prises en compte pour établir le vecteur caractéristique de chaque région, donc de dimension 12.

Comme décrit précédemment, l'évolution des différents paramètres, entre l'image originale et les images filtrées passe-bas, est prise en compte. Chaque paramètre est ainsi non pas absolu mais relatif.

1.3.3 Approche fréquentielle

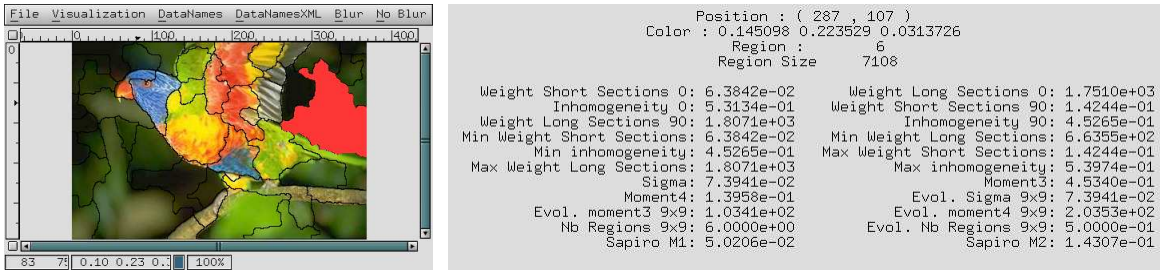
Le rehaussement et la restauration d'images bruitées ou floutées fournissent un grand nombre d'algorithmes[Centeno et Haertel, 1997, Deguchi *et al.*, 2002, Kennedy et Basu, 2000] capables d'apporter une certaine discrimination pour notre reconnaissance de flou. Nous avons retenu l'algorithme de l'approche de Sapiro[Sapiro et Ringach, 1996] dont la méthode est particulièrement efficace en cela qu'elle rehausse (visuellement) le contraste localement avec un très bon respect général des différents éléments de l'image. Il s'agit en fait d'un algorithme de diffusion anisotropique couleur, algorithme itératif dont le temps de traitement est relativement conséquent. Néanmoins il est possible d'accélérer considérablement cet algorithme comme montré dans [Colantoni *et al.*, 2003]. Ainsi les effets d'un rehaussement de ce type devrait avoir un effet plus prononcé sur les régions floues, évolution que nous allons quantifier. Au final, les premiers moments statistiques sont calculés pour chaque région, à la fois pour l'image initiale et l'image rehaussée.

1.4 Partie expérimentale

1.4.1 Ensemble référence

Avant d'appliquer un algorithme d'apprentissage, il faut construire un ensemble de régions floues et non floues. De plus, les différents paramètres doivent être calculés sur chacune de ces régions. La première étape est donc la constitution d'une base d'images généralistes dans lesquelles des parties floues apparaissent. Le choix de ces images s'est fait afin de respecter une grande diversité de contexte ou de types de flou.

Ensuite, nous avons mis en place un outil graphique. Comme illustré à la figure 1.7, l'interface présente à l'utilisateur le résultat de la segmentation. Les utilisateurs doivent alors cliquer sur une région puis choisir l'option "flou" ou "non flou". Automatiquement les divers paramètres sont calculés sur la région et ceux-ci sont insérés dans l'ensemble référence avec l'option retenue.



(a) Choix de la région (b) Informations sur la région

Fig. 1.7 – *Interface manuelle de sélection des régions floues et non floues.*

1.4.2 Algorithmes de classification

Après le passage de cette sélection manuelle, nous obtenons une liste de vecteurs numériques classifiés en deux types : "flous" et "non flous". Il devient donc nécessaire d'introduire des algorithmes d'apprentissage automatique qui vont permettre d'établir un modèle sur la base préalable pour ensuite réaliser une classification automatique "flou"/"non flou" sur n'importe quelle région. L'apprentissage par l'exemple nous offre un large panel de possibilités. Toutefois, il n'existe pas de cheminement générique qui permette de choisir le meilleur classifieur pour une tâche précise [Mitchell, 1997]. C'est pour cette dernière raison que nous avons testé un certain nombre d'algorithmes de classification afin de sélectionner le mieux adapté. Présentons tout d'abord les approches potentiellement exploitables.

- Décision bayésienne
  Il s'agit simplement d'appliquer la règle de bayes au vecteur caractéristique. Plus précisément, il faut calculer pour chaque paramètre la probabilité P("flou") que la région soit floue [Da Rugna *et al.*, 1997]. Une étape de discrétisation automatique préalable est donc nécessaire, étape réalisée par la méthode FUSINTER[Rabaseda-Loudcher *et al.*, 1996].
- Arbres de décision
  Dans le but de réaliser une classification, les méthodes basées sur des règles sont les plus naturelles. On entend par règles une propriété induite par une condition, par exemple : "SI variable <valeur ALORS propriété". De nombreuses techniques[Duda *et al.*, 2001] aboutissent à la création de règles à partir d'exemples classifiés. On peut citer les méthodes par

induction de règles (CN2) ou encore les arbres de décision. Ces derniers sont une approximation d'une valeur discrète par partitionnement de l'espace d'observation des exemples. Une feuille de l'arbre représente une partition (depuis la racine jusqu'à la feuille) d'individus similaires. Dans une feuille, les individus sont considérés comme appartenant à une même classe: la majoritaire. En d'autre termes, rechercher la classe d'un individu inconnu revient à rechercher la feuille dans l'arbre pour cet individu. La classe qui lui sera attribuée sera celle de sa feuille. Le passage de l'arbre à une série de règles est direct: il faut suivre le partitionnement de la racine à la feuille. Cela donne des règles de type :

$$\begin{aligned} Si \quad & Valeur1 < x1 \quad et \\ Si \quad & Valeur2 < x2 \quad et \\ Si \quad & Valeur3 < x3 \quad alors \\ & ClasseJ \end{aligned}$$

Cette méthode est intelligible: l'expert est capable de comprendre et d'analyser les règles produites ainsi que leur fonctionnement. Le choix de la méthode proprement dite, celle de génération de l'arbre, est guidé par la volonté de compromis entre, d'un coté, l'obtention de résultats satisfaisants et, de l'autre, une certaine restriction quant à la taille de l'arbre. En effet, plus l'arbre sera étendu, plus le nombre de règles augmente et moins l'expert sera à même d'appréhender le classifieur dans sa globalité. La prolifération des règles perturbe en fait la lisibilité de la méthode. Ainsi, face à ID-3 ou CART notamment, nous avons retenu la méthode C4.5[Quinlan, 1996b].

- Réseau de neurones
  Classiquement, les réseaux de neurones apportent, dans une grande majorité de cas, un pouvoir discriminant non négligeable. Il est donc logique de tester ces approches dans la reconnaissance de flou. En revanche, contrairement aux méthodes par règles, ce type de classifieur est très abscon pour l'utilisateur. Le système rend en effet une décision sans qu'il soit possible d'interpréter le cheminement qui a conduit à cette décision. Un réseau classique avec back-propagation et niveaux cachés va permettre d'apprécier au mieux l'apport possible de l'approche neuronale : ce réseau allie simplicité et généricité tout en produisant des résultats probants. Le nombre de niveaux cachés ou encore la fonction de transfert ont été configurés afin d'obtenir les meilleurs résultats. Naturellement, une attention toute particulière a été portée sur le paramétrage, dans l'objectif de ne pas tomber dans le piège de la sur-adaptation(ou sur-apprentissage): le réseau connaîtrait "par coeur" les exemples et ne saurait donc plus extrapoler ses connaissances à des individus inconnus. Pratiquement, nous avons utilisé l'outil Weka[Garner, 1995] quant à l'implémentation du réseau

de neurones.

- Support Vector Machine
  Les dernières avancées dans la théorie de l'apprentissage ont donné lieu à un outil habituellement performant : les Machines à Vecteurs Support[Vapnik, 1995] (Support Vector Machine - SVM). Il s'agit d'un outil moderne de classification efficace sur de nombreux problèmes, spécifiquement dans le cas de données non séparables. Nous avons donc implémenté, via la librairie SvmLight[Joachims, 1999], les SVM afin de les tester dans la reconnaissance du flou. Signalons de plus que les SVM ne souffrent pas en principe du travers de la sur-adaptation.
- Boosting, Bagging
  Les méthodes par boosting[Schapire, 1999] et par bagging[Quinlan, 1996a] ont vu le jour afin d'améliorer l'efficacité d'un classifieur, quel qu'il soit. Le boosting peut être vu comme une combinaison d'un ensemble de classifieurs. Ceux-ci sont obtenus par la même méthode d'apprentissage mais basés sur différentes conjectures sur l'espace des exemples. Similairement, le bagging est un "bootstrap": le classifieur est appliqué itérativement sur un sous-ensemble aléatoire de l'espace des exemples. C'est cette dernière approche[Breiman, 1996] que nous avons retenue afin d'améliorer les différents classifieurs cités.

1.4.3 Évaluation

L'évaluation de notre méthode d'extraction de flou va se baser sur des critères basiques du monde de l'apprentissage automatique. Plus précisément, après avoir entraîné le classifieur sur une série d'exemples il s'agit de juger de sa capacité à discriminer des exemples inédits. Commençons donc par introduire quelques notions simples nous permettant de juger des résultats, en intégrant le fait que nous possédons une vérité terrain obtenue manuellement.

1.4.3.1 Mesures d'efficacité

Soit N le nombre d'éléments à classifier, f_i la classe décidée par le classifieur et a_i la classe réelle du i-ème élément. On considèrera le flou comme une réponse positive.

- Efficacité - Mesure le rapport entre le nombre de régions correctement décidées et le nombre de régions totales, c'est-à-dire le taux de reconnaissance global.

$$Efficacite = \frac{\|\{i; f_i = a_i\}\|}{N} \tag{1.2}$$

- Précision - Mesure le rapport entre le nombre de régions floues reconnues à juste titre et le nombre de régions floues décidées. Plus précisément la précision permet de juger si la

méthode ne décide pas trop souvent à tort qu'une région est floue.

$$Precision = \frac{\|\{i; f_i = a_i \& a_i = true\}\|}{\|\{i; f_i = true\}\|} \tag{1.3}$$

- Rappel - Mesure le rapport entre le nombre de régions floues reconnues à juste titre et le nombre réel de régions floues. Plus précisément le rappel permet de juger si la méthode "n'oublie" pas trop de régions floues.

$$Rappel = \frac{\|\{i; f_i = a_i \& a_i = true\}\|}{\|\{i; a_i = true\}\|} \tag{1.4}$$

- Faux positifs (FP) - Mesure le pourcentage de régions classées à tort floues. Ce taux permet de juger si la méthode n'a pas tendance à classer trop souvent "flou" des regions. C'est un critère important dans notre étude. En effet, nous désirons détecter des régions floues et les marquer dans une optique indexation d'images. Pour cela, il est nécessaire que les régions dites "floues" le soient réellement. Le taux de faux positif doit donc être le plus bas possible.

$$FP = \frac{\|\{i; f_i = true \& a_i = false\}\|}{\|\{i; a_i = false\}\|} \tag{1.5}$$

1.4.3.2 La validation croisée

L'évaluation d'une méthode par validation croisée[Kohavi, 1995] est un standard. Nous avons choisi une validation croisée par pas de 10. Plus précisément, à chacun des 10 tours, les exemples sont partagés en deux groupes: 90% servant pour l'apprentissage et 10% servant pour la validation. Les mesures d'efficacité sont alors mesurées sur les 10% de validation, la moyenne des valeurs fournissant le résultat final. Cette méthode d'évalution possède les qualités attendues pour notre étude:

- Non biaisée - Les résultats obtenus ne privilégient pas une méthode ou une autre ni ne privilégient une certaine classe d'exemples.
- Faible variance - Répéter le test donnera des résultats similaires, même avec peu d'exemples.
- Gestion du sur-apprentissage - Les différents tirages aléatoires permettent de s'assurer que les résultats d'efficacité ne sont pas dus au phénomène de sur-apprentissage.

1.4.4 Résultats

Les résultats que nous présentons ici sont obtenus sur la même base d'images. Cette dernière inclut 50 images différentes et ainsi 1000 régions floues et non floues.

Les tables 1.1 et 1.2 donnent les résultats obtenus avec une segmentation de type WaterShed tandis que le tableau 1.3 se propose de comparer la méthode WaterShed et la méthode pyramidale. La méthode Bagging sur les réseaux de neurones n'a pas été mise en pratique pour une

raison simple: la complexité d'un apprentissage par réseau de neurones est telle qu'il n'est pas possible de réaliser un bagging en un temps raisonnable. Deux types de résultats sont présentés :

- Tous les paramètres - La méthode d'apprentissage automatique est lancée sur l'ensemble des paramètres de régions. Ainsi la taille du vecteur résultant est de 80.
- Sélection de paramètres - La méthode d'apprentissage automatique est lancée sur un sous-ensemble des paramètres de régions. Uniquement les paramètres les plus corrélés avec l'information "flou/non flou" sont retenus, soit une dizaine d'entre eux formant le vecteur. Nous reviendrons plus en détail sur cette sélection en aval.

Methode	C4.5	Bagging C4.5	ANN	SVM
Effi cacité	80.7%	88.3%	85.1%	68.8%
Précision	82.1%	89.1%	87.5%	81.6%
Rappel	82.8%	89.7%	89.3%	51.7%
FP	18%	10.5%	19%	14.0%

Tab. 1.1 – *Validation croisée: Tous les paramètres*

Methode	Bayes	Bagging bayes	C4.5	Bagging C4.5	ANN	Bagging ANN	SVM
Effi cacité	75.1%	79.2%	84.3%	88.0%	85.3%	88.4%	61.0%
Precision	77.0%	80.1%%	86.7%	88.64%	87.8%	89.5%	80.2%
Rappel	71.1%	72.1%	82.5%	88.35%	84.4%	88.0%	38.2%
FP	15%	12.1%	13.1%	9.8%	12.12%	10.2%	14.1%

Tab. 1.2 – *Validation croisée: Sélection de paramètres*

L'efficacité des méthodes est globalement meilleure dans le cas de la sélection de paramètres. Cela s'explique simplement par le fait que de nombreux paramètres sont corrélés engendrant par cela une certaine sur-adaptation pour les classifieurs. Ainsi la classification Bayesienne n'a pas été appliquée dans le cas où l'ensemble des paramètres est choisi, pour la raison simple que la corrélation inter-paramètres est trop forte. Mais avant toute chose, revenons sur l'analyse des descripteurs et la sélection qui en découle. Pour illustrer notre propos sur un cas restreint, la figure 1.8 illustre la correlation sur les descripteurs correspondant aux min et max des 3 paramètres issus des matrices de longueur de plages calculées sur l'ensemble du cadran. Cette étude de correlation permet de se rendre compte que certains descripteurs sont très corrélés comme "Min Poids petites sections" et "Max Poids petites sections" et d'autres moins comme "Max homogénéité" et "Min Poids grandes sections". Ainsi, par sélection de paramètres peu corrélés entre eux il est possible d'extraire un vecteur caractéristique de degré moindre (15 variables dans notre exemple) tout

aussi discriminant que le vecteur pris dans sa globalité. Ainsi l'extraction d'un nombre plus faibles de paramètres conduit à une extraction plus rapide.

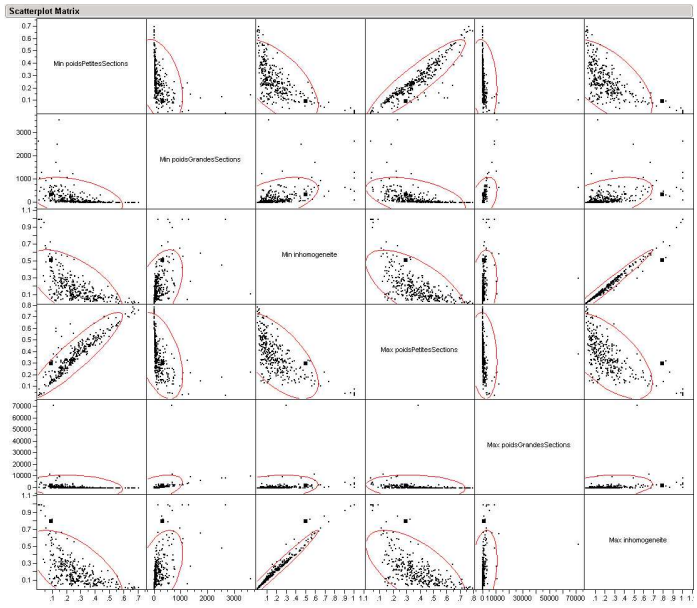


Fig. 1.8 – *Correlation entre descripteurs*

À part la méthode par machines à vecteurs support, les méthodes donnent des résulats malgré tout très similaires. Le taux de faux positif FP est d'environ 10%, ce qui est très raisonnable et encourageant. Comme l'illustre la figure 1.10 via une segmentation WaterShed où les régions floues sont noircies, la détection est visuellement probante. Ces images sont obtenues avec la méthode C4.5 sans bagging, en utilisant la sélection de paramètres. Naturellement les images montrées ne font pas partie de l'ensemble d'apprentissage. C4.5 semble dans notre contexte la méthode proposant le meilleur compromis entre des résultats satisfaisants, le temps de calcul, et, finalement, la compréhension de la décision.

Un autre intérêt immédiat d'une méthode de type C4.5 est sa capacité à donner une sorte de "qualité" sur la décision rendue. En effet, une fois que la feuille de l'arbre correspondant à un inconnu est trouvée, on affecte à cet inconnu la classe majoritaire de la feuille. Un principe simple serait qu'une feuille de 100 éléments dont 99 seraient de la classe A serait alors beaucoup plus pertinente qu'une feuille de 20 éléments dont seulement 11 seraient de la classe A. Ainsi, il est possible de mettre un seuil de qualité à la détection afin de réduire le taux de faux positif. Ce sont les images de cette détection avec précision maximale qui sont présentées à la figure1.10. Ici un taux minimum d'une majorité de 90% de flou dans une feuille est appliqué pour décider de classer une région en floue.

Les résultats décevants de la méthode par machines de vecteurs support sont certes surprenants mais doivent s'expliquer par le fait que cette approche n'est pas adaptée aux types de données de ce problème qui sans doute ne sont pas séparables par des hyperplans.

Methode	C4.5 + WaterShed	C4.5 + Pyramide	ANN + WaterShed	ANN + Pyramide
Precision	86.7%	83.1%	87.8%	84.3%
Rappel	82.5%	80%	84.4%	82%

Tab. 1.3 – *Validation croisée: Segmentation via WaterShed et Pyramide*

Le tableau 1.3 montre les résultats obtenus avec la méthode C4.5 classique suivant les deux méthodes de segmentation, WaterShed et Pyramide sur deux classifieurs. Globalement la méthode pyramidale donne des résultats très similaires mais légèrement moins satisfaisants que la méthode WaterShed. Ainsi les algorithmes C4.5 et WaterShed forment le couple le plus convaincant de cette étude. La figure 1.9 illustre dans ce cadre un arbre de décision obtenu par l'algorithme C4.5.

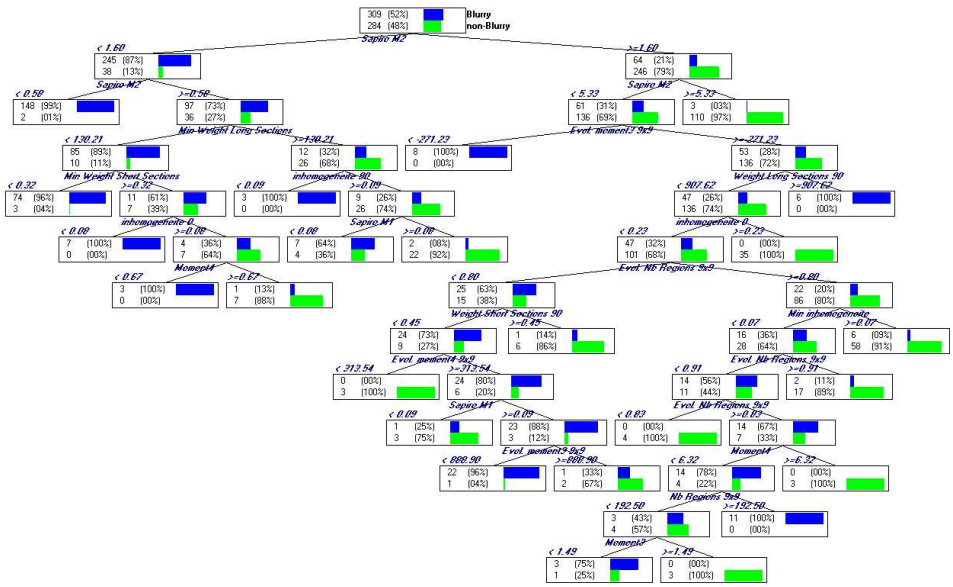


Fig. 1.9 – *Exemple de classification C4.5*

1.5 Perspectives

Nous avons proposé une méthode pour marquer une image d'une méta-donnée importante: la localisation des zones floues. L'application en indexation d'images, même si non implémentée

ici, est évidente. Traiter tous les pixels de l'image équitablement peut ne pas être judicieux si certains appartiennent à une région floue. En effet, ces dernières n'ont pas la même information intrinsèque que les autres régions: au premier abord, les parties non floues émergentes captent l'attention du spectateur. Les résultats obtenus sont relativement satisfaisants en cela qu'ils fournissent un bon taux de reconnaissance. Néanmoins notre approche est pour l'instant encore naïve, différents points doivent être améliorés:

Tout d'abord l'utilisation d'autre descripteurs s'impose. On pense notament à des descripteurs basés sur des transformées en ondelettes[Jain et Healy, 1998]. De part les propriétés de ces transformations une meilleure détermination du flou devrait être alors possible. D'autre part, le réglage et le choix de la méthode de segmentation doit permettre de diminuer le nombre de régions mixtes, ie des régions possédant des zones floues et nettes. Il conviendrait d'utiliser des critères spécifiques pour contraindre la méthode de segmentation à ne pas mélanger les zones nettes et les zones floues. Par exemple, on pourrait intégrer à la méthode un classifieur simple qui, en cas de doute sur une région, imposerait que cette région soit re-segmentée. Ainsi il deviendra possible d'obtenir une segmentation optimisée pour une détection du flou.

À ce niveau de l'étude qui plus est, nous répondons par Oui ou Non à la question "est-ce flou ?" Or ce n'est peut être pas une décision si tranchée qui convient à tous les cas de figure. Un continuum entre le flou et le net permettrait d'affiner l'information extraite. Dans ce cadre, chaque région ne serait pas définie par un booléen de type 0 ou 1 mais par une valeur variant de 0 (net) à 1 (flou) exprimant la notion du degré de flou estimé. Ainsi, l'information "plus ou moins flou" pourrait être considérée dans une méta-donnée.

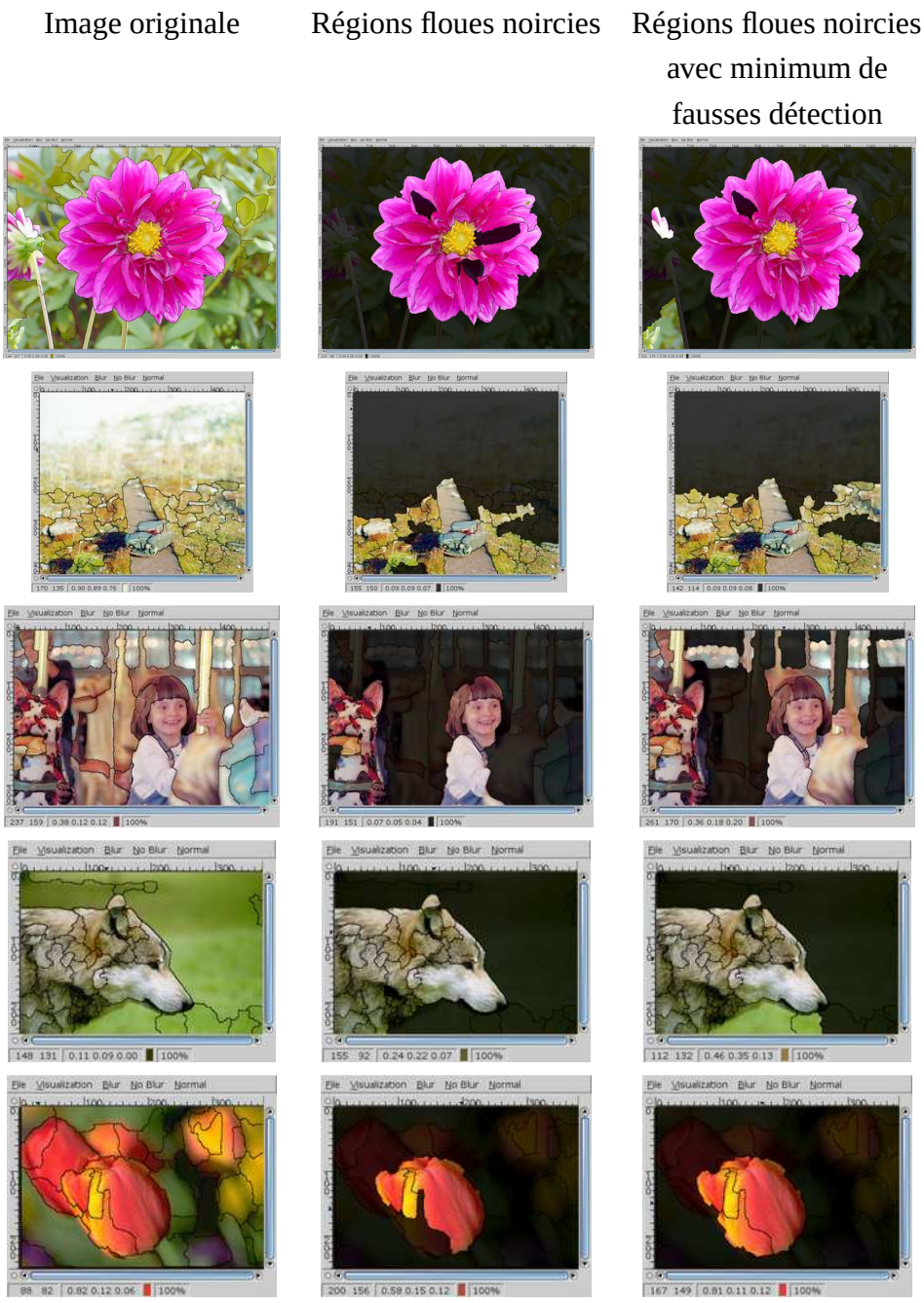


Fig. 1.10 – *Exemples d'extraction de zones floues*

Chapitre 2

SÉLECTION COULEUR BAS NIVEAU DE RÉGIONS

Sommaire

L'optimisme placé dans les outils de segmentation doit être mesuré, telle pourrait être la conclusion rapide de la partie précédente. Néanmoins, en prenant en compte les caractéristiques d'une segmentation grossière issue de l'approche pyramidale, peut t-on transformer cette segmentation en données numériques stables sans se fourvoyer sur les capacités réelles de l'outil. Ainsi, dans une optique annoncée de pré-filtrage, nous proposons une mesure de similarité basée sur une sélection couleur de régions dans une image. À partir d'une segmentation, des régions sont sélectionnées dans le plan Teinte-Valeur. Finalement, des distances sont proposées afin de réaliser la mesure de similarité finale entre deux groupes de régions.

Comme précédemment évoqué lors de la première partie, la tentative d'extraction de l'information "sémantique" à partir d'une segmentation n'est que trop souvent illusoire. D'autant plus que la méthode de segmentation ne restera que relative et qu'aucun algorithme ne répondra comme le ferait le système humain. En revanche, nous avons introduit un outil pyramidal, qui bien qu'imparfait, permet d'obtenir des régions homogènes tant au niveau couleur que texture. Ainsi, en restant dans un cadre d'utilisation de méthodes bas niveaux pour la recherche d'images par le contenu, est-il possible d'extraire un vecteur numérique caractéristique de l'image à partir de cette segmentation ? Et plus spécifiquement, dans le contexte de l'apparence et l'impression couleur rendue par une image. Se posera alors également le problème de construire des distances adaptées à la mesure de similarité colorimétrique entre deux populations. En définitive, dans ce contexte applicatif de filtrage rapide mais local et intelligible pour l'utilisateur, plusieurs règles guident cette étude :

- La segmentation doit être grossière.
- L'extraction de paramètres numériques à partir des régions est non sémantique.
- Seule l'information couleur est prise en compte.

Évidemment, dans la continuité des conclusions précédemment évoquées dans la partie I, nous sommes loin ici des attentes exprimées de l'utilisateur. Mais cette étape doit être vue comme un outil "complémentaire" dans un moteur de recherche. Nous nous plaçons notamment dans une optique de pré-filtrage de la base complète où ne seraient sélectionnées que les images potentiellement intéressantes, en terme de sensation couleur, pour focaliser la recherche future.

Dans ce contexte, le choix de la méthode pyramidale semble justifié à la vue des objectifs. Cette méthode a en effet été préalablement étudiée, montrant sa tendance à ne pas regrouper des formes trop importantes. De plus, afin d'accentuer plus encore cette propriété, la version un germe par pixel du sommet est choisie dans cette étude. Ainsi les partitions sont complexes mais conservent l'information couleur de l'image considérée, comme l'illustre à nouveau la figure 2.1 sur quelques exemples. Les pixels colorimétriquement éloignés ne seront pas a priori regroupés. Par contre, la segmentation reste très loin de ce que pourrait souhaiter visuellement un utilisateur...

Dans ce cadre, nous commencerons par présenter la méthode de sélection automatique des régions représentatives de l'image avant de pouvoir les caractériser et les composer via des distances de similarité adaptées. Afin de critiquer et valider l'approche, il est par contre impossible de se raccrocher à un vérité terrain. Seules les techniques de visualisation basées sur les classiques planches de résultats, vont nous permettre d'illustrer notre approche [1].

1. Ce travail à fait l'objet de deux publications [Da Rugna et Konik, 2002a, Da Rugna et Konik, 2003]

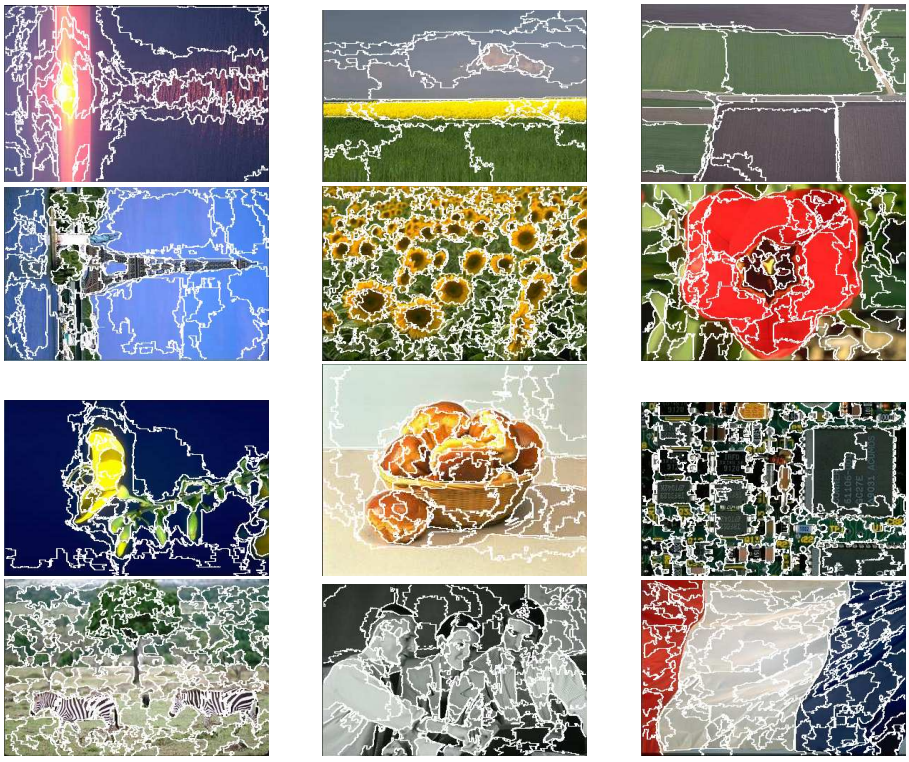


Fig. 2.1 – *Quelques exemples de segmentation “grossière”.*

2.1 Sélection couleur de régions

À partir de cet ensemble de régions, nous proposons une méthode afin d'extraire les régions les plus représentatives de l'image. Pour cela, pour toutes les images, un nombre pré-défini C de classes est fixé par l'utilisateur final. L'algorithme de segmentation fournit pour chaque région les deux caractéristiques suivantes :

- Sa moyenne dans l'espace {R,G,B}
- Sa dispersion couleur ΔE {L*,a*,b*}

En excluant toute autre source d'information, comme l'information spatiale, il convient alors d'exploiter au mieux l'information couleur. Dans un premier temps, nous déterminons C classes de régions qui permettront ensuite d'en extraire les représentants optimaux. Deux possibilités nous ont semblé appropriées à cette séparation en classes : soit l'utilisation de la dispersion couleur ΔE, soit l'utilisation de l'espace HSV [2] et plus précisément de la composante H. En effet, notre objectif est de réaliser une sélection offrant un compromis entre qualité et rapidité. Une sélection sur un seul axe est ainsi plus adaptée à cet objectif. Ensuite, la composante H permet de séparer les régions en classes de teintes similaires ce qui, dans une optique filtrage par le contenu, semble cohérent. Utiliser ΔE correspondrait en effet à rassembler les régions similaires, d'un point de vue dispersion colorimétrique, ce qui semble a priori moins souhaitable.

Ainsi, chaque région est caractérisée par la moyenne HSV de ses pixels. La composante couleur H permet de définir une vision circulaire de la teinte de 0 à 360 degrés: le rouge est aussi bien 0 que 359. En utilisant une distance adaptée, ie qui prend en compte la circularité de H, et un algorithme de type nuées dynamiques, C classes de régions sont ainsi extraites, comme le montre la figure 2.2.

Finalement, dans chaque classe, deux régions sont sélectionnées: les valeurs min et max de la composante V. Ces deux régions représentent les deux régions les plus attractives de la classe partant de l'axiome que la focalisation se fait sur les régions les plus lumineuses et/ou les plus sombres. La figure 2.3 illustre, sur quelques exemples, les régions ainsi sélectionnées, au nombre de $N = 2 \times C$, où C est fixé à 5 a priori. Les régions obtenues sont très diverses tant en taille,en forme ou en position dans l'image. De plus, comme tout seuil fixé a priori, la réponse reste satisfaisante si l'on suppose que le seuil puisse véritablement contenir toutes les couleurs significatives de l'image. Lorsque le nombres de couleurs de l'image reste inférieure à ce seuil, les régions obtenues sont cohérentes avec l'information couleur. En revanche, certaines couleurs ne sont pas prises en compte, comme le rouge dans l'image du panier d'oeufs de Pâques par

2. La transformation de RGB à HSV est réalisée par l'algorithme décrit dans la première partie de ce document sur la représentation de la couleur

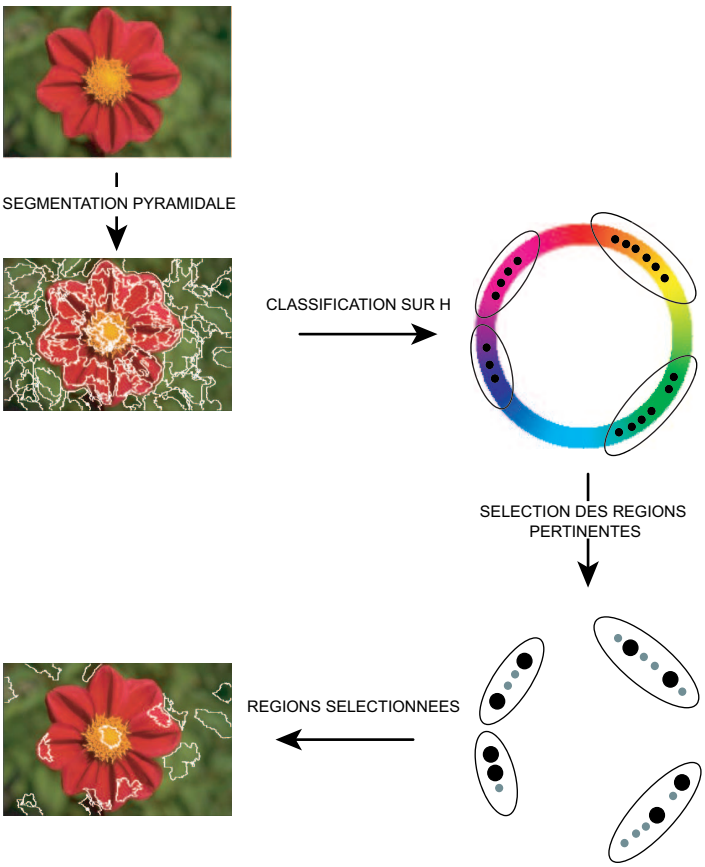


Fig. 2.2 – *Extraction des régions émergentes par la teinte*

exemple. Mais le but reste principalement de trouver les régions représentatives des couleurs de l'image.

Il est sans doute aisé de faire un parallèle entre cette approche et les approches de quantification et d'extraction de couleurs caractéristiques. Cependant, l'étape de segmentation pyramidale permet tout d'abord de supprimer le bruit et les petits objets mais aussi permet de regrouper spatialement des couleurs. Ainsi, par rapport à une quantification sur le nuage complet, seul l'aspect coloré de la région entière est pris en compte, et non celui de chaque pixel. Les textures sont ainsi résumées en une seule couleur, la moyenne de la région considérée.

2.2 Extraction de connaissances et recherche de similarité

Nous avons proposé une nouvelle méthode pour extraire un ensemble de N régions homogènes d'une image couleur, suffisamment représentatives de l'information visuelle que cette dernière véhicule. Mais deux questions se posent alors: comment décrire numériquement les régions obtenues d'une part, et comment ensuite comparer deux ensembles de régions afin d'appréhender

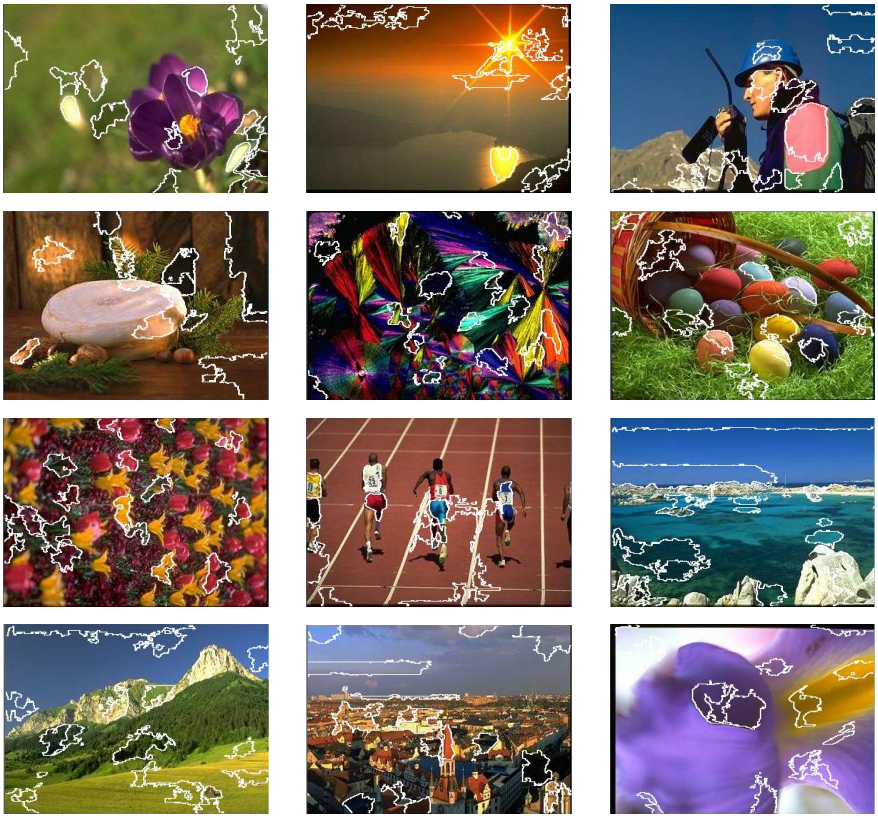


Fig. 2.3 – *Quelques exemples de sélection “couleur”*

une certaine similarité entre images.

2.2.1 Descripteurs

La première étape est ainsi de décrire numériquement chaque région extraite. Nous l'avons vu précédemment, trois types de mesures peuvent être effectuées sur une région: sa couleur, sa forme et sa texture. Néanmoins, afin de rester cohérent avec notre sélection effectuée sur le seul critère couleur, cette unique information va être conservée.

Ainsi chaque région est décrite par:

- Sa moyenne dans l'espace {R,G,B}
- Sa dispersion couleur ΔE {L*,a*,b*}

Dans une approche pratique, l'utilisateur sélectionne une image cible I. Le moteur interroge alors la base de paramètres où sont stockés les descripteurs α_i^I pour chaque région $i \in [1..N]$. Une mesure de similarité adaptée doit être maintenant introduite pour mesurer la distance entre deux ensembles de régions sachant qu'aucune relation d'ordre ne régit ces ensembles.

2.2.2 Mesures de similarité

Il s'agit donc de quantifier l'écart entre deux images I et J représentées par deux vecteurs de descripteurs $(\alpha_1^I,...,\alpha_n^I)$ et $(\alpha_1^J,...,\alpha_n^J)$, où aucune relation ne lie la k^{e} région de l'image I et celle de l'image J. Bien-sûr, il existe diverses distances mathématiques pour le réaliser mais elles ne correspondent pas à l'objectif recherché. Ici un appariement de régions est nécessaire[Lew, 2001], d'où cette première et classique version:

Algorithme 3: Algorithme de base de l'appariement

Données : Deux ensembles de régions $(\alpha_1^I,...,\alpha_n^I)$ et $(\alpha_1^J,...,\alpha_n^J)$

pour chaque $r \in [1...n]$ **faire**

Rechercher k_r et l_r tel que:

$$d_r = d(\alpha_{k_r}^I,\alpha_{l_r}^J) = \min_{\forall k \neq k_1,k_2,...,k_{r-1},\forall l \neq l_1,l_2,...,l_{r-1}} \{d(\alpha_k^I,\alpha_l^J)\}$$

$$d(I,J) = \sum_{r=1}^{n} d_r;$$

Résultat : Mesure de similarité $d(I,J)$

Considèrons ensuite les régions issues de la segmentation de deux images comme les sommets d'un graphe non orienté, valué, complet et biparti. Si le poids de l'arête joignant deux régions vaut la distance euclidienne entre elles, alors cette distance somme l'ensemble des poids du couplage minimal de ce graphe.

En fonction de la similarité recherchée, cette distance peut être aisément modifiée par les approches suivantes:

- mettre un seuil d'acceptation qui rejette deux régions trop différentes. Ceci permet de limiter le calcul de distance à des régions visuellement suffisamment proches.
  L'algorithme devient alors l'algorithme 4

Algorithme 4: Élimination par seuil

Données : Deux ensembles de régions $(\alpha_1^I,...,\alpha_n^I)$ et $(\alpha_1^J,...,\alpha_n^J)$
Un critère d'arrêt Seuil

pour chaque $r \in [1...n]$ **faire**
  Rechercher k_r et l_r tel que:

$$d_r = d(\alpha_{k_r}^I,\alpha_{l_r}^J) = \min_{\forall k \neq k_1,k_2,...,k_{r-1},\forall l \neq l_1,l_2,...,l_{r-1}} \{d(\alpha_k^I,\alpha_l^J)\}$$

$r_{max} = n$;
pour chaque $r \in [1...n]$ **faire**
  si $d_r > Seuil$ **alors**
    $r_{max} = Min(r,r_{max})$

$$d(I,J) = \sum_{r=1}^{r_{max}} d_r;$$

Résultat : Mesure de similarité $d(I,J)$

- éliminer de façon arbitraire les couples les moins proches. On apparie alors un nombre plus restreint de régions en ignorant ainsi pour chaque image une partie de celle-ci. L'algorithme devient alors l'algorithme 5 où les 2 derniers couples (par exemple) sont exclus de l'appariement.
- ajouter un poids inversement proportionnel à l'ordre obtenu via le couplage minimal. Les couples les plus éloignés sont présents dans le calcul de la distance mais leur dissimilarité est pondérée face aux couples les plus proches. En fait, la mesure se calcule alors suivant l'algorithme 6.

Algorithme 5: Élimination arbitraire

Données : Deux ensembles de régions $(\alpha_1^I,...,\alpha_n^I)$ et $(\alpha_1^J,...,\alpha_n^J)$

pour chaque $r \in [1...n-2]$ **faire**

Rechercher k_r et l_r tel que:

$$d_r = d(\alpha_{k_r}^I,\alpha_{l_r}^J) = \min_{\forall k \neq k_1,k_2,...,k_{r-1},\forall l \neq l_1,l_2,...,l_{r-1}} \{d(\alpha_k^I,\alpha_l^J)\}$$

$d(I,J) = \sum_{r=1}^{n-2} d_r$;

Résultat : Mesure de similarité $d(I,J)$

Algorithme 6: Appariement pondéré

Données : Deux ensembles de régions $(\alpha_1^I,...,\alpha_n^I)$ et $(\alpha_1^J,...,\alpha_n^J)$
Une pondération $\beta_1,....,\beta_n$

pour chaque $r \in [1...n]$ **faire**

Rechercher k_r et l_r tel que:

$$d_r = d(\alpha_{k_r}^I,\alpha_{l_r}^J) = \min_{\forall k \neq k_1,k_2,...,k_{r-1},\forall l \neq l_1,l_2,...,l_{r-1}} \{d(\alpha_k^I,\alpha_l^J)\}$$

d(I,J) = $\sum_{r=1}^{n} \beta_r \times d_r$

Résultat : Mesure de similarité $d(I,J)$

Ces trois approches se justifient pleinement, chacune donnant son propre type de réponses. Calculer l'appariement complet laisse supposer que deux images que l'on juge similaires ont exactement les mêmes régions colorimétriquement représentatives. Or, si l'on considère les deux images de la figure 2.4, il est clair que cette supposition n'est pas correcte. L'image de droite va engendrer des régions dans les troncs et le sol qu'on ne retrouvera pas dans l'image de gauche. Pourtant, la présence commune du même arbuste impliquerait une certaine similarité.

Fig. 2.4 – *Deux ensembles de régions proches*

La figure 2.5 présente des exemples obtenus dans l'outil ***i***COBRA, où l'influence du choix de la distance est sensible. Comme l'illustrent ces quelques exemples de recherche, il est possible d'extrapoler certaines tendances, même s'il est délicat de juger leur pertinence dans l'absolu. La distance avec élimination arbitraire fait donc matcher une partie seulement des couleurs de l'image. On peut imaginer l'appliquer si l'on veut par exemple rechercher des similarités dans différents contextes. Comme classiquement, la distance par seuil donne naissance à un comportement plus instable, un seuil préfixé ne pouvant pas s'adapter à toutes sortes d'images. Les distances basées sur l'appariement de base ou pondérée sont assez similaires, à ceci près que la pondération privilégie les images dont les premiers appariements sont faibles. En effet, la pondération renforce l'influence des régions colorimétriquement très proches. Chacune de ces distances va donc trouver son application en fonction de l'attente que l'on aura du pré-filtrage ou de la similarité recherchée.

Finalement, à la figure 2.6, nous proposons les classiques planches de similarité, de gauche à droite, de haut en bas. La distance pondérée est utilisée pour obtenir ces planches.

Les résultats obtenus semblent globalement correspondre à nos attentes, bien que seule l'information couleur sur les régions "de focalisation" soit prise en compte et que la subjectivité de l'utilisateur soit mise à l'épreuve. Néanmoins, il semble délicat de comparer nos résultats avec d'autres moteurs de recherche par le contenu. En effet, seule une validation subjective est possible pour appréhender la pertinence des résultats.

(a) Appariement de base

(b) Appariement de base

(c) Appariement par seuil

(d) Appariement par seuil

(e) Élimination arbitraire

(f) Élimination arbitraire

(g) Appariement pondéré

(h) Appariement pondéré

Fig. 2.5 – *Influence de la distance sur la recherche*

2.3 Conclusion

Nous avons présenté une méthode de séléction de régions basée uniquement sur l'élément perceptif couleur. Cette méthode repose sur une segmentation grossière de l'image combinant les informations spatiales et colorimétriques via l'usage d'une pyramide Gaussienne. À l'issue, nous avons proposé une technique de sélection automatique des régions principales sans intervention de l'utilisateur. Cette sélection bas-niveau, sans notion de sémantique, permet d'extraire des régions significatives au sens couleur de l'image considérée. Parallèlement au classique schéma "segmentation, région, paramètres" nous proposons donc de réduire le nombre de région à un sous ensemble réduit.

Néanmoins, le problème de la distance adéquate étant posé, nous avons introduit et étudié différentes distances adaptatives afin de créer une mesure de similarité. Les résultats, certes graphiques, prouvent qu'une utilisation de cet outil ne serait pas fortuit. En effet, dans une optique

Fig. 2.6 – *Recherche de similarité par sélection de régions*

de pré-filtrage notament, la similiarité que nous avons fait émerger peut être utilisée. Plus précisément, dans l'hypothèse où on désire écarter de la recherche des images qui ne seront pas solutions, il s'agit d'écarter les images qui ne sont pas proches au sens de notre similarité. En effet, une image dont les régions représentatives ne sont pas relativement proches de celles de l'image requête, peut être considérée dans certain cas comme trop éloignée pour devenir solution potentielle. Et, considérant cette problèmatique, la stabilité et la rapidité de la méthode plaident pour son utilisation.

Chapitre 3

IMPLÉMENTATION D'UNE ALGÈBRE D'HISTOGRAMMES

Sommaire

Fédérer toutes les caractéristiques d'une image en un modèle permettant de mieux appréhender la sémantique de celle-ci est devenu un enjeu important. Dans ce contexte nous proposons une algèbre d'histogrammes permettant d'écrire des requêtes a priori complexes dans un langage simple. À partir du langage AQL, Array Query Language, nous montrerons qu'il est ainsi possible d'appliquer ce langage dans le cadre de recherches d'images par histogrammes.

3.1 Introduction

Jusqu'à présent, nous nous sommes concentrés sur l'application de la similarité dans des requêtes par le contenu des images. Pour cela, nous avons approfondi les limites et les contraintes imposées par des méthodes bas-niveaux depuis la segmentation jusqu'à la mise en place de distances. Le but de ce chapitre, né lors de mon séjour de 6 mois à San Diego dans le laboratoire SDSC sous la direction des professeurs Amarnath Gupta et Simone Santini, est de jeter les bases d'un langage de requêtes adaptable à tout outil maîtrisé du traitement d'images. En effet, dans les systèmes de recherche d'images par le contenu, les données sont archivées usuellement dans une base de données relationnelles. Ainsi, via le classique langage SQL, il est possible d'effectuer des requêtes sur cette base. Néanmoins, l'algèbre relationnelle sur laquelle se base le langage SQL n'est sans doute pas adaptée aux requêtes que l'on désire réalisées. "Images database are not database with images" est un des axiomes reconnus par tous... Plus précisément, si à chaque image correspond une ligne de descripteurs numériques, qu'en est-il ensuite de la manière de traduire une requête de l'utilisateur en requête SQL ? Comment intégrer le modèle de connaissance sur les données dans la manière d'exécuter la requête proprement dite ? Différentes voies ont été proposées[Müller *et al.*, 2003, Santini, 2001] afin de contourner les limites de l'algèbre relationnelle base du SQL. À partir de ce constat, Santini et Gupta[Santini et Gupta, 2001, Santini et Gupta, 2000] proposent une algèbre de paramètres à même de prendre en compte les caractéristiques des paramètres.

Dans ce contexte, se pencher sur le cas de la recherche à partir d'histogrammes trouve un sens. En effet, bien que cette approche soit très populaire, il n'existe pas de langage permettant d'écrire des requêtes spécifiques à cet outil. Plus précisément, la communauté propose des distances spécifiques mais non un langage générique pour utiliser pleinement l'information contenue dans l'histogramme. Or, un histogramme n'est pas uniquement une série de données mais bel et bien une information intelligible et accessible. À partir du travail de Libkin[Libkin *et al.*, 1996] sur un langage de requête pour les tableaux multidimensionnels, nous proposons ici une implémentation d'une algèbre sur histogramme et illustrerons dans le langage mis en place diverses requêtes de recherche d'images. Durant tout ce chapitre, plutôt qu'un formalisme pointu [1], nous avons opté pour une présentation que nous pouvons qualifier de "par l'exemple", notre objectif étant de faire comprendre au lecteur simplement les modèles mis en jeu.

1. Formalisme détaillé dans les articles [Libkin *et al.*, 1996] et [Santini et Gupta, 2000]

3.2 Limites de la modélisation algébrique usuelle

Généralement, dans les moteurs de recherches d'images par le contenu, la phase "hors ligne" se résume en quatre points:

- Extraction des données numériques à partir de chaque image. Ces données peuvent être mono-dimensionnelles, la moyenne couleur par exemple ou multi-dimensionnelles, la matrice de longueur de plages par exemple.
- Enregistrement dans une base de données. Les données numériques sont enregistrées dans la base pour chaque image.
- Insertion dans la base de données d'une ou plusieurs distances à même de réaliser la mesure de similarité entre deux images.
- Indexation des données suivant un critère précis. Afin de faciliter la recherche d'images par la suite, les données sont indexées afin d'établir des arbres de recherche notamment.

La phase "en ligne" correspond alors à écrire en langage SQL la requête proprement dite. Dans ce modèle, chaque descripteur est donc vu comme une entité fermée de données : il est possible de calculer une distance entre deux descripteurs mais on ne peut pas interagir avec le contenu même du descripteur. Or les nombreux paramètres complexes que l'on peut extraire d'une image modélisent souvent bien plus qu'une simple série de nombres. Il est notament possible de moduler un ou plusieurs descripteurs entre eux pour en former des nouveaux.

Prenons un exemple simpliste mais très démonstratif. Si l'on possède les données d'un histogramme de 256 couleurs, n'est-il pas possible de transformer celui-ci en histogramme de 16 couleurs ? Et ceci, bien sûr sans refaire le calcul sur l'image d'origine. Or, transformer directement par ré-échantillonnage un histogramme de 256 bins en 16 bins est tout à fait justifiable dans des cas spécifiques. Si on souhaite uniquement appréhender le contenu couleur prédominant d'une image, on peut par exemple limiter le calcul de la distance sur deux populations de taille réduite. Néanmoins, imaginer cette requête dans le langage SQL devient alors très difficile. Il est certes possible de la réaliser, mais il s'agit plus alors d'une stratégie au coup par coup, c'est-à-dire sans modèle valide et concret. Modèle qui permettrait d'explorer toutes les possibilités offertes par le decripteur tout en conservant une grande capacité d'optimisation. Limiter l'utilisateur au modèle relationnel classique bride "l'expressivité" [2] même du descripteur. Ainsi, une base de données "images" se doit d'intégrer un langage permettant de considérer les caractéristiques propres des divers descripteurs.

2. Les auteurs parlent ici de "Expressiveness gap" entre les données et le modèle permettant leur gestion

3.3 Langage de requête pour les tableaux multidimensionnels

Libkin [Libkin *et al.*, 1996], non convaincu du modèle SQL dans de nombreuses applications, propose le AQL, Array Query Language, permettant d'exprimer simplement des requêtes complexes sur des données de type tableaux multidimensionnels. Reprenons ici l'exemple donné par les auteurs pour introduire le langage AQL. Supposons que l'on possède les informations suivantes:

- T, un tableau mono-dimensionnel contenant pour le mois de Juin les températures relevées toutes les heures à la surface.
- RH, un tableau mono-dimensionnel contenant pour le mois de Juin les taux d'humidité relevés toutes les heures à la surface.
- WS, un tableau bi-dimensionnel contenant pour le mois de juin la vitesse du vent relevée toutes les demi-heures à différentes altitudes.

Ainsi, en supposant que la notion de chaleur insupportable est fonction de la température, du taux d'humidité et de la vitesse du vent, comment répondre à la question : "Quels jours de Juin furent insupportablement chauds ?" bien sûr il est possible de répondre à cette question en langage SQL, mais la requête qui en découlera sera très compliquée et la recherche qui en résultera probablement peu optimisée. En langage AQL, la requête s'exprime ainsi :

```
{d \mid \d <- gen!30,
         (* pour chaque jour *)

      \WS' == evenpos!(proj_col!(WS,0)),
          (* Ajuste la dimension de WS *)

      \TRW == zip_3!(T,RH,WS'),
          (* Combine les données *)

      \A == subseq!(TRW,d*24,d*24+23),
          (* extraire la donnée pour le jour d *)

      heatindex!(A) > threshold};
          (* Filtre pour insupportablement chaud *)
```

Sans entrer dans le détail de l'implémentation ou du schéma de construction du langage AQL nous allons présenter maintenant quelques fonctions du langage, puis les principales règles de celui-ci.

3.3.1 Les fonctions proposées

On note $\langle e \mid i_1 < e_1,...,i_k < e_k \rangle$ un tableau multidimensionnel de dimension k tel que e_j soit le nombre d'éléments de la j^{e} dimension. $\langle e \mid i < 5 \rangle$ est donc un tableau mono-dimensionnel de longueur 5.

La dimension $dim_k(e)$ est le n-uplet composé du nombre d'éléments des k dimensions de e. $dim_{j,k}(e)$ correspond au nombre d'éléments de la j^{e} dimension de e. Par souci de simplification, dans le cas de tableaux mono-dimensionnels, $dim_{1,1}$ est noté $len(e)$.

Ainsi les fonctions suivantes sont proposées:

$$
\begin{aligned}
map\ f\ A &= \langle\ f(A[i]) \mid i < len(A)\ \rangle \\
zip\ A\ B &= \langle\ A[i],B[i] \mid i < min(len(A),len(B))\ \rangle \\
subseq\ A\ i\ j &= \langle\ A[i+k] \mid k < (j+1) - i\ \rangle \\
reverse\ A &= \langle\ A[len(A) - i - 1] \mid i < len(A)\ \rangle \\
evenpos\ A &= \langle\ A[i*2] \mid i < \frac{len(A)}{2}\ \rangle \\
transpose\ A &= \langle\ A[i,j] \mid j < dim_{2,2}(A),i < dim_{1,2}(A)\ \rangle \\
proj\_col\ A\ j &= \langle\ A[i,j] \mid i < dim_{1,2}(A)\ \rangle \\
multiply\ A\ B &= \langle\ \sum\{A[i,j] * B[j,k] \mid j < dim_{2,2}(A)\} \mid i < dim_{1,2}(A),k < dim_{2,2}(B)\ \rangle
\end{aligned}
$$

3.3.2 Règles de construction

Entrons brièvement dans le détail du langage AQL. Trois règles de construction sont proposées.

Tout d'abord, un calcul ensembliste. On note $\{e \mid GF_1,...,GF_n\}$ l'ensemble généré et filtré par les règles GF_i. Une règle GF est de type :

1. Génération: $\backslash x \leftarrow A$; x prend toutes les valeurs de A
2. Filtre : $x \in A$; vrai si x est dans l'ensemble A

Ainsi on peut décrire les ensembles suivants :

- $\{x \mid \backslash x \leftarrow A,\ x < 10\}$
  Sous ensemble de A dont les valeurs sont inférieures à 10.
- $\{(x,y) \mid \backslash x \leftarrow A,\ \backslash y \leftarrow B\}$
  $A \times B$
- $\{x \mid \backslash x \leftarrow A,\ x \in B\}$
  $A \cap B$

La deuxième règle proposée est la règle du pattern matching. Prenons des exemples d'utilisation simple.

- $\{(x,y,z) \mid (\backslash x,\backslash y) \leftarrow A,(y,\backslash z) \leftarrow B\}$
  Définition de la jointure naturelle entre A et B.
- $\{(x,y) \mid (\backslash x,0,\backslash y) \leftarrow A\}$
  Sélectionne les tuples de A dont la seconde dimension est 0 et projette sur le plan (dimension 0, dimension 2).

Finalement, à la manière d'un langage fonctionnel, il est possible de réaliser des affectations dans des block. Par exemple:

- $val\ \backslash x = 2;$
  x prend la valeur 2.
- $val\ \backslash x = reverse\ A$
  x prend la valeur du tableau inverse de A.
- $val\ \backslash y = \{x \mid \backslash x \leftarrow A, x \in B\}$
  y prend la valeur de $A \cap B$

Nous avons décrit grossièrement le langage par des exemples simples. On trouve la description très précise du langage dans [Libkin *et al.*, 1996]. De plus les auteurs ont implémenté ce dernier et proposé de nombreuses optimisations afin d'accélérer les traitements de la requête. Il s'agit dans ce cas de la stratégie à adopter face à une requête afin de réordonner au mieux les GF, pattern ou block.

3.4 Application aux histogrammes

Un histogramme classique est un tableau mono-dimensionnel. Plus précisément dans notre modèle, un histogramme est de type $\langle e \mid i < N \rangle$ où N est le nombre de couleurs considérées dans l'image. Néanmoins il est facile d'extrapoler cette vision mono-dimensionnelle à une vision multi-dimensionnelle. On peut citer un exemple caractéristique tridimensionnel en réalisant une quantification dans l'espace HSV par un découpage d'abord sur H, ensuite sur S et finalement sur V. Plus globalement, on considère ici comme histogramme tout comptage de pixels qu'il soit réalisé colorimétriquement ou non comme dans le cas d'histogrammes de contours orientés[Nastar *et al.*, 1998].

Nous avons implémenté le langage AQL dans son intégralité ainsi que diverses fonctions adaptées au contexte des histogrammes [3] .

3. L'implémentation est réalisée en Objective Caml avec une partie des optimisations présentées dans [Libkin *et al.*, 1996]

Les autres fonctions suivantes sont ainsi présentes dans l'algèbre finale:

- $Mmax\ A$,$Mmin\ A$
  Retourne le maximun et le mininum d'un tableau mono-dimensionnel A
- add,sub,mul,$div\ldots$
  Opérations élémentaires
- $opendata$
  Ouvre un fichier de données
- $Mop\ operator\ H_1\ H_2$
  Retourne l'ensemble constitué de $operator(H_1,H_2)$*, ie pour des histogrammes mono-dimensionnels:* $\langle\ operator(H_1[i],H_2[i])\ |\ i < Min(len(A),len(B))\ \rangle$
- $Mtraversal\ A\ f$
  Réduit la dimensionalité de A *suivant la fonction* f*, f est donc une fonction de calcul de nouveaux indices. Plus précisément, si B est le résultat de* $Mtraversal\ A\ f$ *alors* $B(f(v)) = A(v)\ \ \forall v$
- $Mproj\ j\ op\ A$
  Projette l'histogramme A *sur la dimension* j *suivant la fonction* op
- $Mdom\ A$
  Renvoi le domaine de l'histogramme A, ie la liste des tailles de chaque dimension

3.5 Exemples de requêtes

Voici des exemples de requêtes qu'il est donc possible d'écrire dans le langage:

- Charger une base d'histogrammes.

```
val \histo = opendata ("histo.data") ;;
```

⇒

```
Opening datafile  of dimension : 6 6 6
0005501.jpg ; 0005502.jpg ; 0005503.jpg ; 0005504.jpg ;
0005505.jpg ; 0005506.jpg ; 0005507.jpg ; 0005508.jpg ;
 ...[size = 400] : [[Float m_array]] bunch
```

- Rechercher les images n'ayant pas plus de 10% de pixels par bins, ie les images qui n'ont pas de couleurs prédominantes.

```
{ x | \x <- histo ,  Mmax(x) < 0.1 } ;;
```

⇒

```
0005504.jpg ; 0005551.jpg ; 0005574.jpg ; 0005778.jpg ;
```

```
0005792.jpg ; 0005793.jpg ; 0005799.jpg ; 0005802.jpg ;
0005832.jpg ; 0005871.jpg ; 0005883.jpg ; 0005924.jpg ;
0005962.jpg ; 0005977.jpg ;  : [[Float m_array]] bunch
```

- Rechercher les images où il y a au moins un pixel dans chaque bin du 1er axe, ie, si le premier axe est un découpage sur la teinte, toutes les images où toutes les teintes sont représentées.

```
{ x | \x <- histo , Mmin(Mproj (1, add , x)) > 0  } ;;
```

⇒

```
0005513.jpg ; 0005519.jpg ; 0005532.jpg ; 0005589.jpg ;
0005832.jpg ;  : [[Float m_array]] bunch
```

- Transformer un histogramme de dimension N en dimension 1. Dans le cas de deux dimensions, le vecteur (i,j) est transformé en i * n + j, n nombre d'éléments de la seconde dimension. Ici la dimension finale sera 6*6*6 soit 216.

```
val \dom = Mdom (element) ;;
val \fun = traversalnto1 (dom) ;;
Mtraversal (element,fun) ;;
```

⇒

```
MARRAY :  [size: 216 ] : Float m_array
```

3.6 Conclusion et perspectives

Le besoin de modèle afin de fédérer les différents descripteurs issus du traitement d'images est grand. Gérer l'ensemble des informations disponibles sur une image ne peut se faire sans un cadre bien précis. Sur ce plan, le modèle classique basé sur le SQL montre ses limites quant à la gestion de bases multimédia. Le pouvoir discriminant des descripteurs est bridé et ne permet pas à l'utilisateur d'explorer toutes les capacités de ceux-ci.

Dans ce contexte, nous avons proposé et implémenté une algèbre sur les histogrammes dérivés d'une algèbre sur les tableaux multi-dimensionnels. Il est ainsi possible d'exprimer des requêtes spécifiques à la problématique des histogrammes dans ce langage et aussi de les exécuter de manière optimale.

Néanmoins, à ce jour, ce travail n'est pas encore abouti, du moins dans sa partie pratique. Deux points méritent ainsi d'être convenablement mis en place. Tout d'abord l'utilisation de ce langage n'est pas satisfaisante, surtout si l'on considère qu'il faut avant tout montrer son utilité. L'étape finale, c'est-à-dire le passage à l'utilisation concrète, n'a pas été faite à ce stade

de l'étude. Il faut donc passer à cette étape en mettant en œuvre ce langage dans un véritable moteur de recherche d'images. En extrapolant le schèma générique d'un système de recherche par le contenu, la figure 3.1 illustre notre propos et les développement futurs. Le principe général reste le même, à ceci près que les requêtes sont écrites dans une algèbre spécifique. Ainsi, les requêtes seront interprétées, optimisées puis exécutées afin de répondre à l'utilisateur. Les différents avantages d'un tel système sont au moins de deux ordres:

- La généricité. Bien sûr, le modèle ne va pas permettre de réaliser des recherches de similarité que l'on ne pourrait faire sur les modèles usuels. Mais dans ce dernier cas, ajouter une nouvelle similarité correspond à ajouter une nouvelle distance dans le moteur de recherche. De par cette vision par entité il est difficile de concevoir un très large nombre de distances. En revanche, dans le modèle algébrique que nous avons proposé, à partir d'un jeu de distances réduit et d'opérations algèbriques simples, il devient possible d'utiliser au mieux toutes les possiblités de recherche offertes par un histogramme.
- Optimisation. Une vision par entité et par cas particuliers d'une recherche de similarité ne permet pas de construire des stratégies d'optimisation optimales face à une requête donnée. Le modèle que l'on propose doit permettre de construire justement ces stratégies en cela que l'on exprime au niveau de la requête elle-même une partie des traitements qui seraient effectués dans le calcul de distance par un modèle classique.

Ensuite, comment utiliser ce modèle algébrique pour fédérer une stratégie de requêtes par l'exemple. En effet, à partir d'une image requête, et donc des descripteurs qui s'y rattachent, comment rechercher dans la base les images espérées ? Plus précisément, comment construire la requête dans l'algèbre proposée ? Plus globalement, le système de gestion des connaissances doit prendre en compte toute l'information portée par les descripteurs et les liens que l'on peut établir entre ceux-ci. Considérer de façon isolée deux descripteurs et leur distance associée aboutit à un système trop fermé pour l'utilisateur. Afin de lui offrir le "Holy Grail", il sera nécessaire de posséder un modèle permettant d'appréhender l'information intrinsèque des descripteurs eux-mêmes et des relations qui les lient.

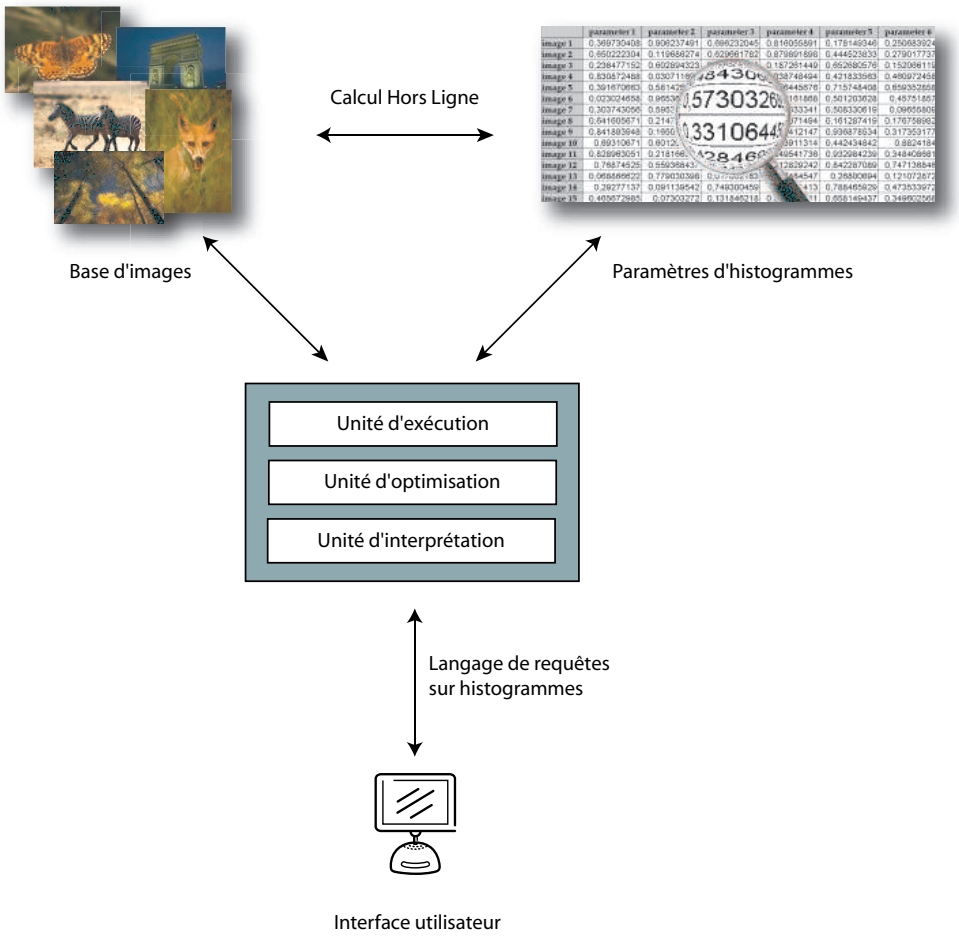


Fig. 3.1 – *Mise en pratique du langage*

CONCLUSION

Réaliser une thèse sur la recherche d'images par le contenu depuis 2000 fut fort exaltant mais aussi fort délicat. En effet, le nombre d'idées, d'approches, d'adaptations liées à l'indexation d'images et la relative jeunesse de ce vaste domaine entraînent quelques contraintes. Notamment, alors que le début de ce travail de recherche était plutôt axé dans une optique de mise en place de descripteurs les plus sémantiques possibles, la constatation logique fut qu'il était illusoire d'espérer appréhender ainsi la complexité d'une base d'images et des attentes de l'utilisateur.

En effet, proposer des planches résultats sur lesquelles la recherche, définie suivant un descripteur, semble fonctionner est trop relatif. La subjectivité de celui qui interroge est fortement mise à contribution, surtout si l'on considère, ce qui est notre cas, l'absence d'applications précises. Néanmoins, notre première approche, étayée par ***i***COBRA, nous a permis d'appréhender les différentes problématiques du domaine. Il n'a pas de toute évidence été possible de retranscrire ici le temps passé à "essayer" des descripteurs, des requêtes. Toutes ces pistes empruntées nous ayant mené à de nombreux passages par des hauts, ie la construction d'un nouveau paramètre, et des bas, ie son application dans nos bases d'images généralistes. À l'écriture de ce manuscrit, il nous a alors semblé plus novateur et constructif d'exposer une vision globale et de ne pas traduire sous forme verbale les efforts passés devant l'écran à tenter d'appréhender la recherche par le contenu.

Toutefois, il nous est possible de proposer des pistes différentes et d'illustrer notre propos par des outils ciblés réalistes face à l'information qu'ils doivent véhiculer. Le condensé que représente cette thèse peut sembler insuffisamment détaillé ou rapide sur certains points, mais nous avons voulu éviter une exhaustivité déplacée.

Contributions apportées

Dans une démarche qui repose sur l'extraction préalable des différentes entités caractéristiques d'une image via des méthodes de bas niveau pour appréhender son contenu, il nous a paru indispensable de quantifier les qualités et défauts des méthodes de segmentation standard. En effet, avant de pouvoir émettre une opinion critique sur les étapes avales menant à la traduction de la similarité, encore faut-il se poser la question de la validité de telle ou telle méthode face à la

réalité fort complexe des images de scènes. Nous avons donc introduit un protocole d'évaluation objectif de la robustesse de ces méthodes face aux contraintes de la généralité et des variations potentielles. La volonté étant de s'assurer qu'il est a priori possible d'obtenir une segmentation d'un objet stable au contexte et aux variations de type changements de teinte, d'illuminant ou autre compression. À partir de cette base tangible, la conclusion serait que, bien qu'il soit toujours possible de segmenter une image, son instabilité interdit malgré tout de lui accorder une confiance somme-toute aveugle. Qui plus est, il devient alors nécessaire de pouvoir pondérer cette confiance pour canaliser les étapes futures.

Bien sûr, cette simple constatation "chiffrée" n'a pas freiné nos ardeurs et nous avons alors axé nos travaux vers d'autres types d'approches plus ponctuelles . Si la segmentation n'est ni stable ni sémantique, on doit se placer dans un contexte approprié où le seul bas niveau est exploitable. Par exemple, nous avons proposé une extraction de régions émergentes sur le seul critère colorimétrique à partir d'une segmentation grossière qui peut trouver son application dans une phase de pré-filtrage.

Ensuite les résultats semblent suffisants pour par exemple collecter ponctuellement des informations ciblées. Dans cette optique, nous avons introduit un détecteur de flou généraliste. L'idée consiste à cumuler un maximum d'informations dites de sémantique induite, afin de mieux adapter la recherche avale par la suite.

En dernier lieu, devant les travers de l'algèbre relationnelle mise à contribution classiquement dans les moteurs de recherches, dont ***i***COBRA, ainsi que l'insuffisante exploitation concrète de descripteurs pourtant intelligibles, sous le couvert de A. Gupta et S. Santini, nous avons jeté les bases et implémenté un langage de requête sur les histogrammes. Certes, ce chapitre laisse au lecteur sans doute plus de perspectives que de conclusions Mais, s'il n'est plus à prouver que les modèles de gestions de connaissances sont un des enjeux futurs du domaine de la recherche de données multimédia, il reste encore à établir les modèles permettant d'assurer cette gestion de manière optimale. Or, l'algèbre proposée peut être une base où l'interrogation directe des descripteurs permettra une complète exploitation des données archivées, sans se heurter à la rigidité du modèle actuel.

Perspectives

Les perspectives les plus actuelles concernent bien évidemment les différents points que nous avons soulevés durant cette thèse. Tous s'appuient sur la volonté de se baser sur des réflexions et des analyses objectives, nous obligeant à définir une base solide et rigoureuse. Ainsi, nos prochains développements vous nous permettre d'approfondir et d'enrichir les résultats présentés ici et de montrer alors la maturité de nos approches.

Plus ponctuellement, reprenons point à point les différentes parties de ce manuscrit afin d'exposer les ouvertures, à court terme, que nous nous devons d'explorer.

- Extraire la meta-donnée "flou" n'est pas en soi une finalité. Il convient maintenant de l'insérer dans un processus de recherche d'images. Améliorer le détecteur sera certes une première étape mais la suivante sera de montrer l'apport de cette extraction. Notamment, dans un schéma de recherche, focaliser les descripteurs sur la seule information "non floue" devrait permettre une meilleure similarité dans un certain type d'images. On pense en particulier à des photographies d'extérieur composées d'objets naturels ou d'animaux, où l'objet est habituellement net alors que le contexte environnant est flou. L'étape finale, ie proposer un modèle complet de gestion du flou, n'est pas véritablement liée à la détection de cet indice visuel mais bien à la mise en place d'un modèle de gestion des connaissances, quelles qu'elles soient.
- L'algèbre que nous avons proposée peut paraître non aboutie mais elle pose des bases qu'il convient maintenant d'enrichir. Actuellement, nous voulons construire et implémenter une algèbre de descripteurs basés sur des ondelettes. Enfin nous intégrerons la gestion des mesures hitogrammes et ondelettes comme base d'un véritable moteur de recherche, et plus précisément nous construirons une interface spécifique globale à même d'interroger la base dans ce langage de requêtes.
- Nous avons étudié les méthodes de segmentation dans un contexte de recherche de similarité, et plus précisément dans dans celui de jugement de la stabilité d'une extraction automatique d'objets. Ce protocole reste certes améliorable sur certains points. Néanmoins, les résultats resteront globalement ce qu'ils sont, c'est-a-dire non significativement convaincants. A fortiori, il devient necessaire à ce point de l'étude de passer ce stade de l'évaluation et de catégoriser sans doute les méthodes de segmentation par rapport à leurs qualités propres. Ainsi, il sera sans doute possible de spécifier, pour une image donnée, définie par une série de meta-données connexes, si une méthode de segmentation sera efficace ou non.

Finalement, si on se replace dans l'objectif initial de cette thèse, la recherche d'images par le contenu, il est clair qu'appréhender la sémantique d'une image ne pourra se faire sans hiérarchiser toutes les informations gravitant autour d'elle, qu'elles soient extraites du contenu de l'image ou encore contextuelles. En effet, les données multimédia ne sont plus uniquement de l'infor-

mation brute. Prendre une photographie avec un appareil numérique permet d'incruster dans le fichier image des informations contextuelles, comme par exemple la focale utilisée, l'utilisation du flash, ou même des mots-clés. Par ailleurs, sortir les images disponibles sur internet de leur contexte, ie les pages où elles sont positionnées, induit une perte d'informations importantes. Dans un cadre général, contraindre l'extraction des données en fonction de pré-connaissances est l'enjeu majeur de notre recherche future. Mais, on se heurte encore à l'absence de modèle conceptuel…

BIBLIOGRAPHIE

[Agnihotri, 1999] A.-M. D. Agnihotri. Mpeg 7: A content description standard beyond compression. Dans I. 42nd Midwest Symposium on Circuits et Systems, éditeurs, *ISCC 99*, 1999.

[Alshatti et Lambert, 1993] W. Alshatti et P. Lambert. Un opérateur optimal pour la détection de contours dans des images couleur. Dans *Actes du $XIV^{èeme}$ Colloque GRETSI*, pages 679–682, Juan-les-Pins (France), 1993.

[Alt et Godau, 1995] H. Alt et M. Godau. Computing the frechet distance between two polygonal curves. *Internat. J. Comput. Geom. Appl.*, 5:75–91, 1995.

[Amadasum et King, 1989] M. Amadasum et R. King. Textural features corresponding to textural properties. *IEEE Transactions On System, Man, and Cybernetics*, SMC-19(5):1264–1274, 1989.

[Andrade *et al.*, 1997] M. C. D. Andrade, G. Bertrand, et A. A. Araújo. Segmentation of microscopic images by flooding simulation: a catchment basins merging algorithm. Dans *Proceedings of SPIE Symposium on Electronic Imaging*, volume 3026, pages 164–175, 1997.

[Ardizzoni *et al.*, 1999] S. Ardizzoni, I. Bartolini, et M. Patella. Windsurf: Region-based image retrieval using wavelets. Dans *DEXA Workshop*, pages 167–173, 1999.

[Asendorf et Hermes, 1996] G. Asendorf et T. Hermes. On textures : An approach for a new abstract description language. *SPIE*, 2657:98–106, 1996.

[Bach *et al.*, 1996] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, et C.-F. Shu. Virage image search engine: An open framework for image management. Dans SPIE, éditeur, *Storage and Retrieval for Image and Video Databases*, pages 76–87, 1996.

[Bachimont, 2003] B. Bachimont. Meaning and indexing: which issues for multimedia documents. Dans *International Workshop on Content-Based Multimedia Indexing (CBMI'2003)*, pages 1–4, 2003.

[Bimbo, 1999] A. D. Bimbo. *Visual information retrieval*. Morgan Kaufmann Publishers Inc., 1999.

[Bimbo *et al.*, 1997] A. D. Bimbo, M. Mugnaini, P. Pala, et F. Turco. Picasso: Visual querying by color perceptive regions. Dans *2nd International Conference on Visual Information Sys-*

tems, pages 125–131, San Diego, 1997.

[Bimbo *et al.*, 1998] A. D. Bimbo, M. Mugnaini, P. Pala, et F. Turco. Visual querying by colour perceptive regions. *Pattern Recognition*, 31:1241–1253, 1998.

[Bimbo *et al.*, 1994] A. D. Bimbo, P. Pala, et S. Santini. Visual image retrieval by elastic deformation of object shapes. *IEEE VL'94, Int. Symp. on Visual Languages*, pages 216–223, 1994.

[Bister *et al.*, 1990] M. Bister, J. Cornelis, et A. Rosenfeld. A critical view of pyramid segmentation algorithms. *Pattern Recognition Letters*, 11:802–809, 1990.

[Blancho et Konik, 2000] P. Blancho et H. Konik. Texture similarity queries and relevance feedback for image retrieval. Dans *International Conference on Pattern Recognition*, volume 4, pages 40–55, 2000.

[Blancho *et al.*, 1998] P. Blancho, H. Konik, et K. Knoblauch. Texture-based similarities between images. *OSA Annual Meeting*, 1998.

[Borsotti *et al.*, 1998] M. Borsotti, P. Campadelli, et R. Schettini. Quantitative evaluation of color image segmentation results. *Pattern Recognition Letters*, 19:741–747, 1998.

[Breiman, 1996] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[Brown et MacAdam, 1949] W. R. J. Brown et D. L. MacAdam. Visual sensitivities to combined chromaticity and luminance differences. *josa*, 39(10):808–834, 1949.

[Burt *et al.*, 1981] P. J. Burt, T. H. Hong, et A. Rosenfeld. Segmentation and estimation of image region properties through cooperative hierarchical computation. *IEEE Trans. Systems Man Cybernetics.*, 12:802–809, 1981.

[Canny, 1986] J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[Cardenas *et al.*, 1993] A. F. Cardenas, I. T. Ieong, R. K. Taira, R. Barker, et C. M. Breant. The knowledge-based object-oriented picquery+ language. *TKDE*, 5(4):644–657, 1993.

[Carson *et al.*, 1999] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, et J. Malik. Blobworld: A system for region-based image indexing and retrieval. Dans *Third International Conference on Visual Information Systems*. Springer, 1999.

[Castelli *et al.*, 1998] V. Castelli, L. D. Bergman, I. Kontoyiannis, C.-S. Li, J. T. Robinson, et J. J. Turek. Progressive search and retrieval in large image archives. *IBM Journal of Research and Development*, 42(2):253–268, 1998.

[Centeno et Haertel, 1997] J. A. S. Centeno et V. Haertel. An adaptive image enhancement algorithm. Dans *Pattern Recognition*, volume 30, pages 1183–1189, 1997.

[Chassery et Garbay, 1984] J. Chassery et C. Garbay. An iterative segmentation method based on a contextual color and shape criterion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6):794–800, 1984.

[Chen *et al.*, 2002] J. Chen, T. N. Pappas, A. Mojsilovic, et B. Rogowitz. Adaptive image segmentation based on color and texture. Dans *Proc. of ICIP 2002*, Rochester, New York, 2002.

[Cheng, 2000] H. Cheng. A hierarchical approach to color image segmentation using homogeneity. *IEEE Trans. on Image Processing*, 9(12):2071–2082, 2000.

[Cheng *et al.*, 2001] H. Cheng, X. Jiang, Y. Sun, et J. Wang. Color image segmentation: advances and prospects. *Pattern Recognition*, 1834(12):2259–2281, 2001.

[Cheng, 1995] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(8):790–799, 1995.

[CIE, 1971] CIE. Colorimetry. Rapport Technique 15, Bureau Central de la CIE, 1971.

[Ciocca et Schettini, 2004] G. Ciocca et R. Schettini. Content-based similarity retrieval of trademarks using relevance feedback. *Pattern Recognition*, 34:1639–1655, 2004.

[Cocquerez et Philipp, 1995] J. Cocquerez et S. Philipp. *Analyse d'images: filtrage et segmentation*. Masson, 1995.

[Colantoni, 2003] P. Colantoni. Color space transformations. Rapport Technique, Laboratoire LIGIV, 2003. http://www.couleur.org.

[Colantoni *et al.*, 2003] P. Colantoni, N. Boukala, et J. Da Rugna. Fast and accurate color image processing using 3d graphics cards. Dans *Vision Modeling anv Visualization, VMV 2003*, pages 383–390, 2003.

[Colantoni et Trémeau, 2003] P. Colantoni et A. Trémeau. 3d visualization of color data to analyze color images. Dans *Proceedings of PICS Conference*, pages 500–505, Rochester, USA, 2003.

[Colombo *et al.*, 1997] C. Colombo, A. Rizzi, et I. Genovesi. Histogram families for color-based retrieval in image databases. Dans *ICIAP (2)*, pages 204–211, Florence, Italy, 1997.

[Comaniciu et Meer, 2001] D. Comaniciu et P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Trans. Pattern Analysis Machine Intelligence*, 24:438–445, 2001.

[Correia et Pereira, 2002] P. Correia et F. Pereira. Standalone objective segmentation quality evaluation. *Journal on Applied Signal Processing*, 2002(4):389–400, 2002.

[Da Rugna *et al.*, 2004] J. Da Rugna, P. Colantoni, et N. Boukala. Hybrid color spaces applied to image database. Dans *Electronic Imaging, SPIE*, volume 5304, pages 254–264, 2004.

[Da Rugna et Konik, 2001] J. Da Rugna et H. Konik. Adaptation du détecteur de harris pour l'indexation de textures. Dans *Gretsi'01 Traitement du Signal et des Images*, 2001.

[Da Rugna et Konik, 2002a] J. Da Rugna et H. Konik. Color coarse segmentation and regions selection for similar images retrieval. Dans *Color in Graphics, Image and Vision*, pages 241–244. Society for Imaging Science and Technology, 2002.

[Da Rugna et Konik, 2002b] J. Da Rugna et H. Konik. Color interest points detector for visual information retrieval. Dans *Electronic Imaging, SPIE*, volume 4672, pages 139–146, 2002.

[Da Rugna et Konik, 2003] J. Da Rugna et H. Konik. Similarity distances evaluation for query by example retrieval. Dans *Electronic Imaging, SPIE*, volume 5018, pages 304–315, 2003.

[Da Rugna et Konik, 2004a] J. Da Rugna et H. Konik. Automatic blur detection for meta-data extraction in content-based retrieval context. Dans *Electronic Imaging, SPIE*, volume 5304, pages 285–295, 2004.

[Da Rugna et Konik, 2004b] J. Da Rugna et H. Konik. étude comparative de méthodes de segmentation dans une approche orientée indexation. Dans *14ème Congrès de Reconnaissance des Formes et Intelligence Artificielle, RFIA 2004*, volume 1, pages 13–20, 2004.

[Da Rugna *et al.*, 1997] J. Da Rugna, S. D. Palma, et D. Zighed. Apprentissage supervisé polythétique : une évaluation. Dans *Cinquièmes rencontres de la société francophone de classification*, 1997.

[Das *et al.*, 1997] M. Das, E. M. Riseman, et B. Draper. Focus: Searching for multi-colored objects in a diverse image database. Dans *Computer Vision and Pattern Recognition*. IEEE Conference, 1997.

[de Andrade *et al.*, 1999] M. de Andrade, G. Bertrand, et A. Araújo. An attribute-based image segmentation method. *Materials Research*, 2(3):145–151, 1999.

[Deguchi *et al.*, 2002] K. Deguchi, T. Izumitani, et H. Hontani. Detection and enhancement of line structures in an image by anisotropic diffusion. Dans *Pattern Recognition Letters*, volume 23, pages 1399–1405, 2002.

[Deng et Manjunath, 2001] Y. Deng et B. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:800–810, 2001.

[Dombre, 2003] J. Dombre. *Systèmes de représentation multi-échelles pour l'indexation et la restauration d'archives médiévales couleur*. PhD thesis, Université de Poitiers, 2003.

[Duda *et al.*, 2001] R. Duda, P. Hart, et D. Stork. *Pattern Classification (Second Edition)*. Wiley-Interscience, 2001.

[Egenhofer, 1997] M. J. Egenhofer. Query processing in spatial-query-by-sketch. *Journal of Visual Languages and Computing*, 8(4):403–424, 1997.

[Finlayson et Süsstrunk, 2002] G. Finlayson et S. Süsstrunk. Color ratios and chromatic adaptation. Dans *Proc. IST CGIV*, pages 7–10, Poitiers, France, 2002.

[Flickner *et al.*, 1995] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, et P. Janker. Query by image and video content: the qbic system. *IEEE Computer*, 28(9):310–315, 1995.

[Freeman, 1974] H. Freeman. Computer processing of line drawing images. *Computer surveys*, 6:57–97, 1974.

[Fuertes *et al.*, 2001] J. Fuertes, M. Lucena, N. P. de la Blanca, et J. Chamorro-Martínez. A scheme of colour image retrieval from databases. *Pattern Recognition Letters*, 22:323–337, 2001.

[Fuh *et al.*, 2000] C. Fuh, S. Cho, et K. Essig. Hierarchical color image region segmentation for content-based image retrieval system. *IEEE Transactions on Image Processing*, 9:156–162, 2000.

[Funt et Finlayson, 1995] B. Funt et G. Finlayson. Colour constant colour indexing. *IEEE Transaction on pattern analysis and machine intelligence*, 17(5), 1995.

[Galloway, 1974] M. Galloway. Texture analyss using gray-level run length. Dans *Computer graphics and image processing*, volume 4, pages 172–199, 1974.

[Garner, 1995] S. Garner. Weka: The waikato environment for knowledge analysis. 1995.

[Geman et McClure, 1987] S. Geman et D. McClure. Statistical methods for tomographic image reconstruction. Dans *Bulletin of the International Statistical Institute*, volume LII, pages 5–21, 1987.

[Gevers et Smeulders, 1996] T. Gevers et A. Smeulders. A comparative study of several colour models for colour image invariant retrieval. Dans *First Internaional workshop on image databases and multimedia search*, pages 17–27, Amsterdam, 1996.

[Grau *et al.*, 2004] V. Grau, M. A. Raya, C. Monserrat, M. C. J. Lizandra, et L. Martí-Bonmatí. Hierarchical image segmentation using a correspondence with a tree model. *Pattern Recognition*, 37(1):47–59, 2004.

[Grecu et Lambert, 2001] H. Grecu et P. Lambert. Indexation par descripteurs flous : Application à la recherche d'images. Dans *GRETSI*, volume 2, pages 372–379, 2001.

[Hafner *et al.*, 1995] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, et W. Niblack. Efficient color histogram indexing for quadratic form distance functions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(7):729–736, 1995.

[Haralick, 1979] R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67:786–804, 1979.

[Harris et Stephens, 1988] C. Harris et M. Stephens. A combined corner and edge detector. *4th Alvey Vision Conf. Manchester*, pages 189+, 1988.

[Hu, 1962] M. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on informatio theory*, 8:351–326, 1962.

[Huang *et al.*, 1998] J. Huang, S. R. Kumar, M. Mitra, et W. Zhu. Spatial color indexing and applications. Dans *ICCV*, pages 602–607, 1998.

[Huttenlocher *et al.*, 1993] D. Huttenlocher, D. Klanderman, et A. Rucklige. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

[Jain et Healy, 1998] A. Jain et G. Healy. A multiscale representation including opponent color features for texture recognition. *Image Processing, IEEE Transactions on Image Processing*, 7(1), 1998.

[Joachims, 1999] T. Joachims. Making large-scale svm learning practical. advances in kernel methods - support vector learning. 1999.

[Jolion, 1998] J.-M. Jolion. Indexation d'images : nouvelle problématique ou vieux débat ?. Rapport Technique, LRFV INSA - Lyon, 1998.

[Jolion et Bres, 1999] J.-M. Jolion et S. Bres. Indexation d'images et bruit de codage. *Traitement du Signal*, 15(4):309–320, 1999.

[Jolion et Montanvert, 1992] J. M. Jolion et A. Montanvert. The adaptive pyramid: a framework for 2d image analysis. *CVGIP: Image Underst.*, 55(3):339–348, 1992.

[J.P. Cocquerez, 1995] S. J.P. Cocquerez. *Analyse d'images: filtrage et segmentation*. Masson, 1995.

[Karu *et al.*, 1996] K. Karu, A. K. Jain, et R. M. Bolle. Is there any texture in the image? *Pattern Recognition*, 29(9):1437–1446, 1996.

[Kass *et al.*, 1998] M. Kass, A. Witkin, et D. Terzopoulos. Snakes: active contour models. *International journal of computer vision*, 3(6):321–331, 1998.

[Kennedy et Basu, 2000] L. M. Kennedy et M. Basu. Application of projection pursuit learning to boundary detection and deblurring in images. Dans *Pattern Recognition*, volume 33, pages 2019–2031, 2000.

[Kim et Kim, 2003] J.-B. Kim et H.-J. Kim. Multiresolution-based watersheds for efficient image segmentation. *Pattern Recognition Letters*, 24:473–488, 2003.

[Kohavi, 1995] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. Dans *IJCAI*, pages 1137–1145, 1995.

[Konik, 1994] H. Konik. *Contribution de l'approche pyramidale à la segmentation des images texturées*. PhD thesis, Université Jean Monnet, Saint-Étienne, 1994.

[Kreyss *et al.*, 1997] J. Kreyss, M. Röper, P. Alshuth, T. Hermes, et O. Herzog. Video retrieval by still image analysis with imageminer. Dans *Science and Technologie*, San Jose, 1997. SPIE '97.

[Lambert et Grecu, 2003] P. Lambert et H. Grecu. A quick and coarse color image segmentation. Dans *ICIP*, pages 14–17, 2003.

[Lambert et Macaire, 2000] P. Lambert et L. Macaire. Filtering and segmentation: the specifity of colour images. Dans *CGIP'2000 : 1st International Conference on Color in Graphics and Image Processing*, pages 57–71, Saint-Etienne, France, 2000.

[Lau et Levine, 2002] H. Lau et M. Levine. Finding a small number of regions in an image using low-level features. *Pattern Recognition*, 35:2323–2339, 2002.

[Lee et Cok, 1991] H. C. Lee et D. Cok. Detection boundaries in a vector field. *IEEE Trans. on Signal Processing*, 39(5), 1991.

[Lee *et al.*, 1994] J.-H. Lee, B.-H. Chang, et S.-D. Kim. Comparison of colour transformations for image segmentation. *Electronic Letters*, 30(20):1660–1661, 1994.

[Lew, 2001] M. Lew. *Principles of Visual Information Retrieval*. Springer-Verlag, London, 2001.

[Li *et al.*, 1999] Z. Li, O. R. Zaïane, et Z. Tauber. Illumination invariance and object model in content-based image and video retrieval. *Journal of Visual Communication and Image Representation*, 1999.

[Libkin *et al.*, 1996] L. Libkin, R. Machlin, et L. Wong. A query language for multidimensional arrays: design, implementation, and optimization techniques. Dans *Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, pages 228–239, 1996.

[Liew *et al.*, 2001] A. Liew, S. Leung, et W. Lau. Fuzzy image clustering incorporating spatial continuity. *Vision, Image and Signal Processing*, 22:593–601, 2001.

[Linhui et Kitchen, 2000] J. Linhui et L. Kitchen. Object-based image similarity computation using inductive learning of contour-segment relations. *Image Processing, IEEE Transactions on image and video processing for digital libraries*, 9:80–87, 2000.

[Loupias *et al.*, 2000] E. Loupias, N. Seb, S. Bres, et J.-M. Jolion. Wavelet-based salient points for image retrieval. Dans *ICIP*, volume 2, pages 518–521, 2000.

[Lucchese et Mitra, 2001] L. Lucchese et S. Mitra. Color image segmentation: A state-of-the-art survey. *Proceedings of the Indian National Science Academy*, 67(2):207–221, 2001.

[Luo et Guo, 2003] J. Luo et C. Guo. Perceptual grouping of segmented regions in color images. *Pattern Recognition*, 36(12):2781–2792, 2003.

[Ma., 1997] W. Y. Ma.. *NETRA: A Toolbox for Navigating Large Image Databases*. PhD thesis, Dept. of Electrical and Computer Engineering, University of California at Santa Barbara, 1997.

[Ma et Manjunath, 1999a] W.-Y. Ma et B. S. Manjunath. Netra: A toolbox for navigating large image databases. *Multimedia Systems*, 7(3):184–198, 1999.

[Ma et Manjunath, 1999b] W.-Y. Ma et B. S. Manjunath. Netra: A toolbox for navigating large image databases. *Multimedia Systems*, 7(3):184–198, 1999.

[Marfil *et al.*, 2004] R. Marfil, J. A. Rodríguez, A. Bandera, et F. Sandoval. Bounded irregular pyramid: a new structure for color image segmentation. *Pattern Recognition*, 37(3):623–626, 2004.

[Marr, 1982] D. Marr. *Vision: a computational investigation into the human representation and processing of visual information*. W. H. Freeman, San Francisco, 1982.

[Mather et Smith, 2000] G. Mather et D. Smith. Blur discrimination and its relation to blur-mediated depth perception. *Perception*, 31(10):1211–1219, 2000.

[Maxwell et Shafer, 2000] B. Maxwell et S. Shafer. Segmentation and interpretation of multi-colored objects with highlights. *Computer Vision and Image Understanding*, 77:1–24, 2000.

[Mel, 1997] B. Mel. Seemore: combining color, shape, and texture histogramming in a neurally inspired approach to visual object recognition. *Neural Compu*, 9(4):777–804, 1997.

[Mitchell *et al.*, 1977] O. Mitchell, C. Myers, , et W. Boyne. A max-min measure for image texture analysis. *IEEE Transactions on Computers*, 2:408–414, 1977.

[Mitchell, 1997] T. M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[Mokhtarian *et al.*, 1996] F. Mokhtarian, S. Abassi, et J. Kittler. Efficient and robust retrieval by shape content trough curvature scale space. Dans F. international workshop, éditeur, *Image databases and multimedia search*, pages 38–42, 1996.

[Montanvert et Chassery, 1993] A. Montanvert et J.-M. Chassery. *Géométrie discrète en analyse d'images*. Hermès, 1993.

[Mukherjea *et al.*, 1999] S. Mukherjea, K. Hirata, et Y. Hara. Amore: A world wide web image retrieval engine. *The WWW Journal*, 2:115–132, 1999.

[Muller *et al.*, 2001] H. Muller, W. Muller, D. M. Squire, S. Marchand-Maillet, et T. Pun. Performance evaluation in content-based image retrieval: overview and proposals. *Pattern Recognition Letters*, 22:593–601, 2001.

[Munsell, 1946] A. Munsell. *A Color Notation*. Munsell Color Co., Baltimore MD, 1946.

[Müller *et al.*, 2003] H. Müller, A. Geissbuhler, et S. Marchand-Maillet. Extension to the multimedia retrieval markup language: A communication protocol for content-based image retrieval. Dans *European Conference on Content-based Multimedia Indexing (CBMI03)*, Rennes France, 2003.

[Müller, 2001] W. Müller. *Design and implementation of a flexible Content–Based Image Retrieval Framework - The GNU Image Finding Tool*. PhD thesis, Computer Vision and Multimedia Laboratory, University of Geneva, 2001.

[Nastar *et al.*, 1998] C. Nastar, M. Mitschke, C. Meilhac, et N. Boujemaa. Surfimage: A flexible content-based image retrieval system. Dans *ACM International Multimedia Conference*, pages 339–344, 1998.

[Neumann *et al.*, 2002] J. Neumann, H. Samet, et A. Soffer. Integration of local and global shape analysis for logo classification. *Pattern Recognition Letters*, 23(12):1449–1457, 2002.

[Niblack *et al.*, 1993] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glassman, D. Petkovic, et P. Yanker. The qbic project : querying images by content using colour, texture and shape. Dans *SPIE*, volume Storage and retrieval for image and video databases, 1993.

[Novak et Shafer, 1987] C. L. Novak et S. A. Shafer. Color edge detection. Dans *Proc. Of DARPA Image Understanding Workshop*, pages 35–37, Los Angeles (USA). Los Altos, CA, 1987. Morgan Kaufmann Publishers, Inc.

[Ohta *et al.*, 1980] Y. I. Ohta, T. Kanade, et T. Sakai. Color information for region segmentation. *Computer Graphics and Image Processing*, 13:222–241, 1980.

[Ortega *et al.*, 1997] M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, et T. S. Huang. Supporting similarity queries in mars. Dans *5th ACM International Multimedia Conference*, pages 403–413, Seattle, Washington, 1997. ACM.

[Pal et Pal, 1993] N. Pal et S. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26:1277–1294, 1993.

[Park *et al.*, 1998] S. H. Park, I. D. Yun, et S. U. Lee. Color image segmentation based on 3D clustering: morphological approach. *Pattern Recognition*, 31(8):1061–1076, 1998.

[Pass *et al.*, 1996] G. Pass, R. Zabih, et J. Miller. Comparing images using color coherence vectors. Dans *ACM Multimedia*, pages 65–73, 1996.

[Philipp-Foliguet et Lekkat, 2004] S. Philipp-Foliguet et M. Lekkat. Recherche d'images à partir d'une requête partielle utilisant la disposition des régions. Dans RFIA, éditeur, *RFIA*, pages 123–131, 2004.

[Pollefeys *et al.*, 1998] M. Pollefeys, R. Koch, et L. J. V. Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. Dans *ICCV*, pages 90–95, 1998.

[Prewer et Kitchen, 2001] D. Prewer et L. Kitchen. Soft image segmentation by weighted linked pyramid. *Pattern Recognition Letters*, 22:123–132, 2001.

[Quinlan, 1996a] J. R. Quinlan. Bagging, boosting, and c4.5. *Thirteenth National Conference on Artificial Intelligence - Eighth Innovative Applications of Artificial Intelligence Conference*, pages 725–730, 1996. AAAI Press / MIT Press.

[Quinlan, 1996b] J. R. Quinlan. Improved use of continuous attributes in c4.5. *Journal of Artificial Intelligence Research*, 4:77–90, 1996.

[Rabaseda-Loudcher *et al.*, 1996] S. Rabaseda-Loudcher, M. Sebban, et R. Rakotomalala. A comparison of some discretization methods in induction graphs. *Information Sciences : Intelligent Systems*, 1-4(92):137–157, 1996.

[Rao et Lohse, 1989] A. Rao et G. Lohse. Identifying high level features of texture perception. *CVGIP : Graphical Models and Image Processing*, 55(5):218–233, 1989.

[Rao, 1990] A. R. Rao. *A Taxonomy for texture description and identification*. Springer-Verlag, 1990.

[Rezaee *et al.*, 2000] M. Rezaee, P. van der Zwet, B. Lelieveldt, R. van der Geest, et J. Reiber. A multiresolution image segmentation technique based on pyramidal segmentation and fuzzy clustering. *IEEE Transactions on Image Processing*, 9:1238–1248, 2000.

[Roman-Roldan *et al.*, 2001] R. Roman-Roldan, J. Gomez-Lopera, C. Atae-Allah, J. Martinez-Aroza, et P. Luque-Escamilla. A measure of quality for evaluating methods of segmentation and edge detection. *Pattern Recognition*, 34:969–980, 2001.

[Rubner, 1999] Y. Rubner. Perceptual metrics for image database navigation. Rapport Technique CS-TR-99-1621, Stanford University, 1999.

[Sand et Teller, 2004] P. Sand et S. Teller. Video matching. *ACM Trans. Graph.*, 23(3):592–599, 2004.

[Santini, 2001] S. Santini. *Exploratory Image Databases : Content-Based Retrieval*. Academic Press, 2001.

[Santini et Gupta, 2000] S. Santini et A. Gupta. Toward a feature algebra for visual databases: a case study with histogram algebra. Dans VDB-5, éditeur, *Int. Conf. On Visual Databases*, 2000.

[Santini et Gupta, 2001] S. Santini et A. Gupta. Data model for querying wavelet features in image databases. Dans *Multimedia Information Systems*, pages 21–30, 2001.

[Santini et Jain, 1997] S. Santini et R. Jain. Images databases are not databases with images. Dans *9th Int. Conf. on Image Analysis and Process, Lecture Notes in Computer Sciences 1311, Springer*, pages 38–45, 1997.

[Sapiro et Ringach, 1996] G. Sapiro et D. L. Ringach. Anisotropic diffusion of multivalued images to color filtering. Dans *IEEE Transactions on Image Processing*, volume 5, pages 1582–1585, 1996.

[Schapire, 1999] R. E. Schapire. A brief introduction to boosting. Dans *IJCAI*, pages 1401–1406, 1999.

[Schettini, 1993] R. Schettini. A segmentation algorithm for color images. *Pattern Recognition Letters*, 14:499–506, 1993.

[Schiele et Crowley, 1996] B. Schiele et J. L. Crowley. Probabilistic object recognition using multidimensional receptive field histograms. Dans ICPR96, éditeur, *International Conference on Pattern Recognition*, 1996.

[Schmid, 1996] C. Schmid. *Appariement d'images par invariants locaux de niveaux de gris*. PhD thesis, Institut National Polytechnique de Grenoble, 1996.

[Sciascio *et al.*, 1999] E. D. Sciascio, G. Mingolla, et M. Mongiello. Content-based image retrieval over the web using query by sketch and relevance feedback. *Lecture Notes in Computer Science*, 1614, 1999. Springer.

[Shaffrey *et al.*, 2002] C. W. Shaffrey, I. H. Jermyn, et N. G. Kingsbury. Psychovisual evaluation of image segmentation algorithms. Dans *Proceedings of Advanced Concepts for Intelligent Visual Systems*, Ghent, Belgium, 2002.

[Smeulders *et al.*, 2001] A. Smeulders, M. Worring, S. Santini, A. Gupta, et R. Jain. Image databases at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1), 2001.

[Smith et Chang, 1995] J. R. Smith et S.-F. Chang. Automated image retrieval using color and texture. Rapport Technique CU/CTR 408-95-14, CTR, Columbia University, rue General Dufour, 24, CH-1211, Geneva, Switzerland, 1995.

[Smith et Chang, 1996] J. R. Smith et S.-F. Chang. Visualseek: A fully automated content-based image query system. Dans *ACM Multimedia*, pages 87–98, 1996.

[Smith et Brady, 1997] S. Smith et J. Brady. Susan - a new approach to low level image processing. *Int Journal of Computer Vision*, 23(1):45+, 1997.

[Srihari, 1995] R. K. Srihari. Use of multimedia input in automated image annotation and content-based retrieval. Dans *Conference on Storage and Retrieval Techniques for Image Databases*, pages 249–260, San Jose, 1995. SPIE '95.

[Swain et Ballard, 1991] M. J. Swain et D. H. Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.

[Tamura *et al.*, 1978] H. Tamura, S. Mori, et T. Yamawaki. Textural features corresponding to visual perception. *IEEE Transactions On System, Man, and Cybernetics*, SMC-8:476–473, 1978.

[Tissainayagam et Suter, 2005] P. Tissainayagam et D. Suter. Object tracking in image sequences using point features. *Pattern Recognition*, 38(1):105–113, 2005.

[Tominaga et Wandell, 2002] S. Tominaga et B. A. Wandell. Natural scene-illuminant estimation using the sensor correlation. *Proceedings of the IEEE*, 90:42–56, 2002.

[Torres-Mendez *et al.*, 2000] L. Torres-Mendez, J. C. Ruiz-Suarez, L. E. Sucar, et G. Gomez. Translation, rotation, and scale-invariant object recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 30:125–130, 2000.

[Tremeau et Colantoni, 2000] A. Tremeau et P. Colantoni. Region adjacency graph applied to color image segmentation. *IEEE Trans. on Image Processing*, 9(4):735–744, 2000.

[Trémeau *et al.*, 2004] A. Trémeau, C. Fernandez-Maloigne, et P. Bonton. *Image numérique couleur, De l'acquisition au traitement*. Dunod - Cours et applications, 2004.

[Vandenbroucke *et al.*, 1998] N. Vandenbroucke, L. Macaire, et J. G. Postaire. Color pixels classification in a hybrid color space. Dans *International Conference on Image Processing*, pages 176–180, 1998.

[Vanhamel *et al.*, 2001] I. Vanhamel, I. Pratikakis, et H. Sahli. Hierarchical segmentation using dynamics of multiscale color gradient watershed. Dans *Third International Conference, Scale-Space*, pages 371–379, 2001.

[Vapnik, 1995] V. Vapnik. *The Nature of Statistical Learning Theory*. NY, Springer-Verlag, 1995.

[Veltkamp et Tanase, 2002] R. Veltkamp et M. Tanase. Content-based image retrieval systems: A survey. Rapport Technique, University Utrecht, 2002.

[Vendrig *et al.*, 1999] J. Vendrig, M. Worring, et A. W. M. Smeulders. Filter image browsing: Exploiting interaction in image retrieval. Dans *Computer Vision and Pattern Recognition*, pages 147–154. Third International Conference VISUAL '99, 1999.

[Vertan *et al.*, 2002a] C. Vertan, M. Ciuc, V. Buzuloiu, et C. Fernandez-Maloigne. Compact color-texture run-length description for ornamental stones recognition and indexing. Dans *The Hyperion Scientific Journal*, volume 3A, pages 69–75, 2002.

[Vertan *et al.*, 2002b] C. Vertan, N. Richard, et C. Fernandez-Maloigne. Compact color-texture run-length description for ornamental stones recognition and indexing. Dans M. V. A. in Industrial Inspection X, éditeur, *SPIE*, 2002.

[Vincent et Soille, 1991] L. Vincent et P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:583–598, 1991.

[Wang *et al.*, 2001] J. Z. Wang, J. Li, et G. Wiederhold. Simplicity : Semantics-sensitive integrated matching for picture libraries. *IEEE transactions on pattern analysis and machine intelligence*, 23(9):947–963, 2001.

[Wolf et Jolion, 2003] C. Wolf et J.-M. Jolion. Extraction and recognition of artificial text in multimedia documents. *Pattern Analysis and Applications*, 6(4):309–326, 2003.

[Wu et Chen, 1992] C.-M. Wu et Y.-C. Chen. Statistical feature matrix for texture analysis. *CVGIP : Graphical Models and Image Processing*, 54(5):407–419, 1992.

[Zenzo, 1986] S. D. Zenzo. A note on the gradient of multi-image. *Computer Vision, Graphics, and Image Processing*, 33:116–125, 1986.

[Zhang et Lu, 2001] D. S. Zhang et G. Lu. Content-based shape retrieval using different shape descriptors: A comparative study. Dans *IEEE International Conference on Multimedia and Expo*, pages 317–320, Tokyo, Japan, 2001.

[Zhang, 1996] Y. Zhang. A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29:1335–1346, 1996.

[Zhang, 1997] Y. Zhang. Evaluation and comparison of different segmentation algorithms. *Pattern Recognition Letters*, 18:963–974, 1997.

[Ziliani et Jensen, 1998] F. Ziliani et B. Jensen. Unsupervised segmentation using modified pyramidal linking approach. Dans *IEEE Int. Conf. on Image Proc*, pages 303–308, 1998.

Quatrième partie

Annexes

Chapitre A

*i*COBRA, UN SYSTÈME À BUT PÉDAGOGIQUE

Le système ***i***COBRA, comme indiqué en partie I, fut développé dans le but de mieux appréhender les différents points qui constituent un système de recherche par le contenu. En effet, rien de tel que de maîtriser la globalité de la chaîne, notamment dans un cadre d'évaluation objective. Détaillons maintenant les éléments clés du système cités précédemment :

A.1 Bases d'images

Différentes bases d'images sont proposées à l'utilisateur, dont les suivantes accessibles depuis Internet:

- Goodshoot© - Base d'images généraliste de l'agence de presse goodshoot, constituée de 4500 images environ, de taille restreinte (autour de 400×300) et compressées en JPEG (artefacts induits fort visibles).
- Texture2000, Vistex - Bases de textures en niveaux de gris et couleur.
- Paintings - Bases d'images composées de tableaux numérisés. Le format des 3500 images est du JPEG de qualité 75 avec des tailles d'environ 1000×1000.
- Animals - Base d'images constituées d'animaux pris dans leurs éléments naturels pour la plupart. Le format des 1000 images est aussi du JPEG de qualité 75 avec des tailles d'environ 1000×1000.
- FreeFoto - Base de 40000 images diverses et variées dont l'origine est surtout la photographie amateur. On retrouve ainsi fréquemment des séries d'images d'une même scène. De plus, la qualité de la majeure partie des images et des prises de vue est loin d'atteindre celle du monde professionnel. Cette base est aussi annotée par des mots clés.
- Blur - Petite base constituée par nos soins d'images contenant du flou.

A.2 Indexation

Cette partie permet de choisir un ou plusieurs descripteurs, puis de réaliser une requête par l'exemple. Une série de descripteurs (Pre Filtering) permet de réaliser un préfiltrage, ie sélec-

tionner les N plus proches images de l'image requête. Ensuite une autre série de descripteurs (Classification) permet de réaliser la recherche du plus proche dans les N images. Néanmoins, le pré-filtrage reste désactivé par défaut.

Les différents descripteurs sont tous normalisés. Différentes distances sont possibles pour calculer la similarité entre descripteurs. Néanmoins, le mixage des différents descripteurs se fait via une distance euclidienne classique.

Les figures A.1 et A.2 montrent l'interface dans sa globalité et les différents menus de l'interface.

La figure A.3 illustre une recherche par l'exemple. L'image en haut à gauche est l'image requête. Puis de gauche à droite, de haut en bas, apparaissent les images les plus proches. Les distances entre l'image requête et les images résultats sont affichées pour chaque image.

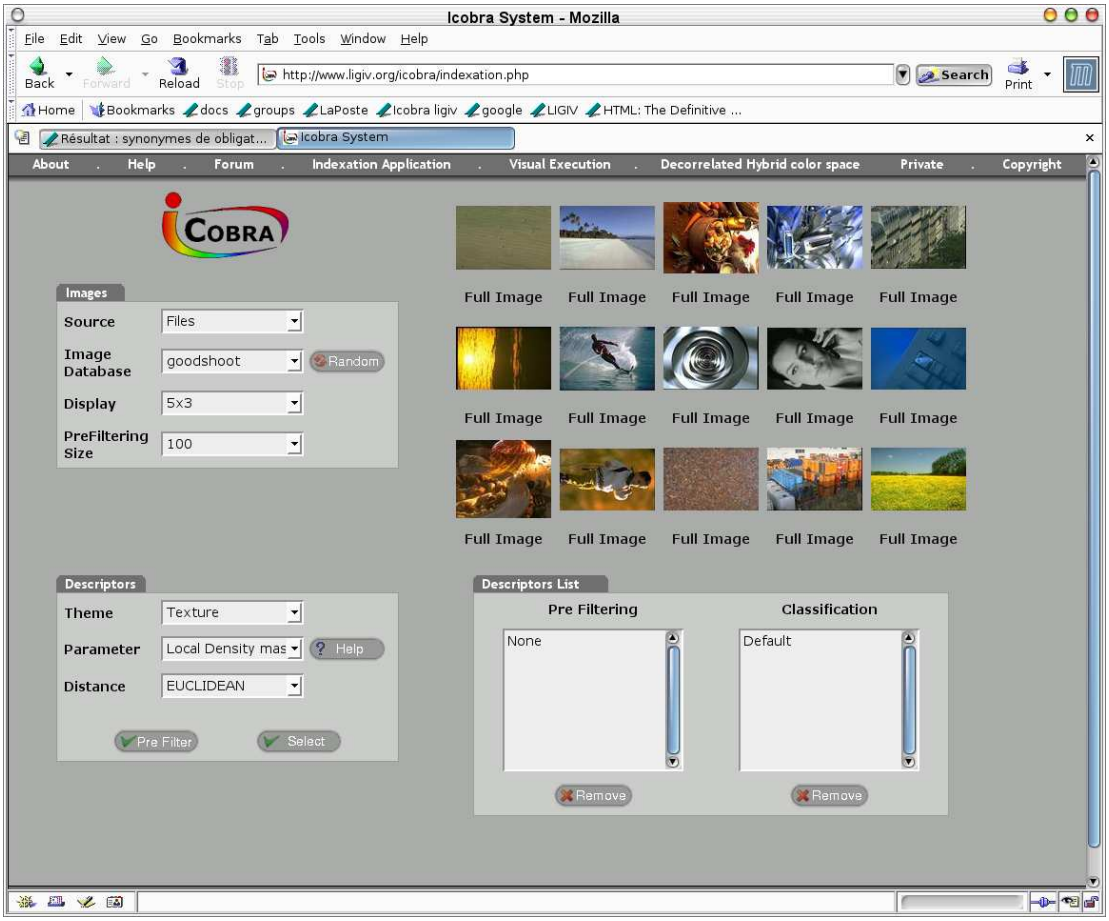


Fig. A.1 – *L'interface d'indexation de* iCOBRA

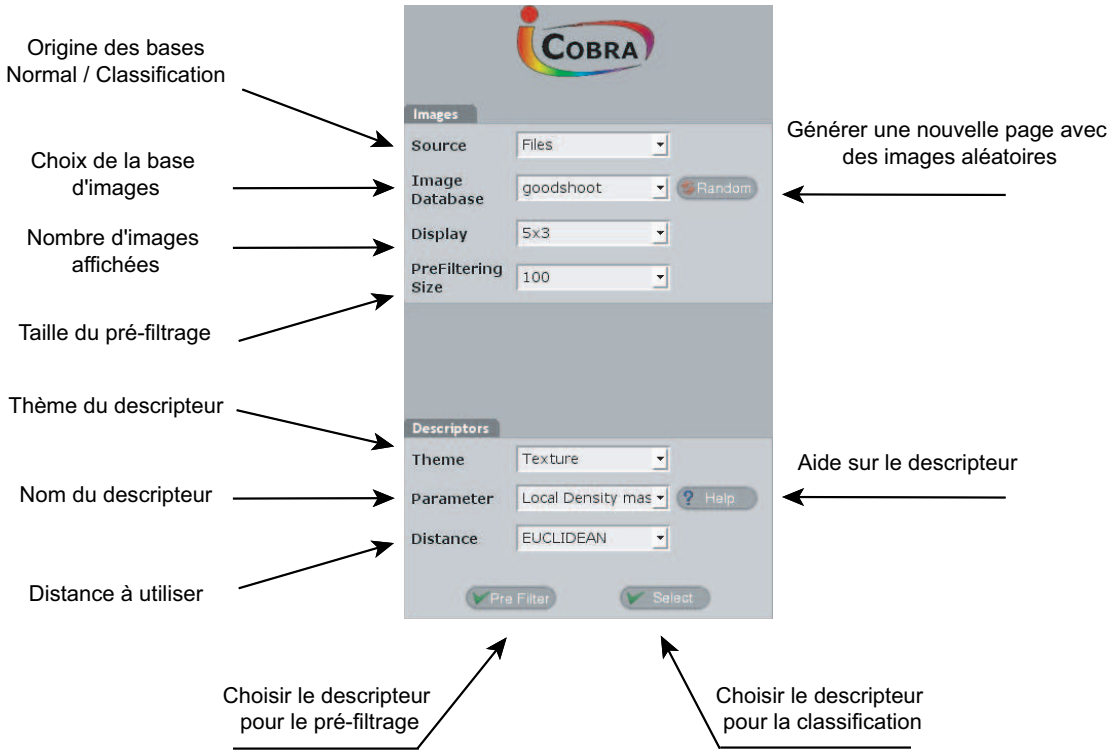


Fig. A.2 – *Exemple de recherche*

A.3 Évaluation de méthodes de recherche

Le principe général de l'évaluation proposée est d'utiliser deux ensembles, l'un que l'on appellera l'ensemble $DATA$, la base de connaissances, et l'autre que l'on appellera l'ensemble $TEST$. Il s'agira ensuite de classer les éléments de $TEST$ ayant connaissance de l'ensemble $DATA$. Pour cela, chaque échantillon est décrit par un vecteur numérique issu de la méthode ou de la combinaison de méthodes que l'on souhaite évaluer. Ainsi, les éléments de $DATA$ sont classés suivant la méthode des k plus proches voisins. Il est donc possible d'établir des taux de bon classement ou des courbes de type Précision/Rappel.

Cette interface de *i*COBRA fut développé en partie pour l'évaluation de descripteurs de textures [Da Rugna et Konik, 2001, Da Rugna et Konik, 2002b]. La figure A.4 représente le schéma aboutissant à la génération des ensembles $DATA$ et $TEST$. Une classe est ainsi définie par toutes les images issue de la même texture.

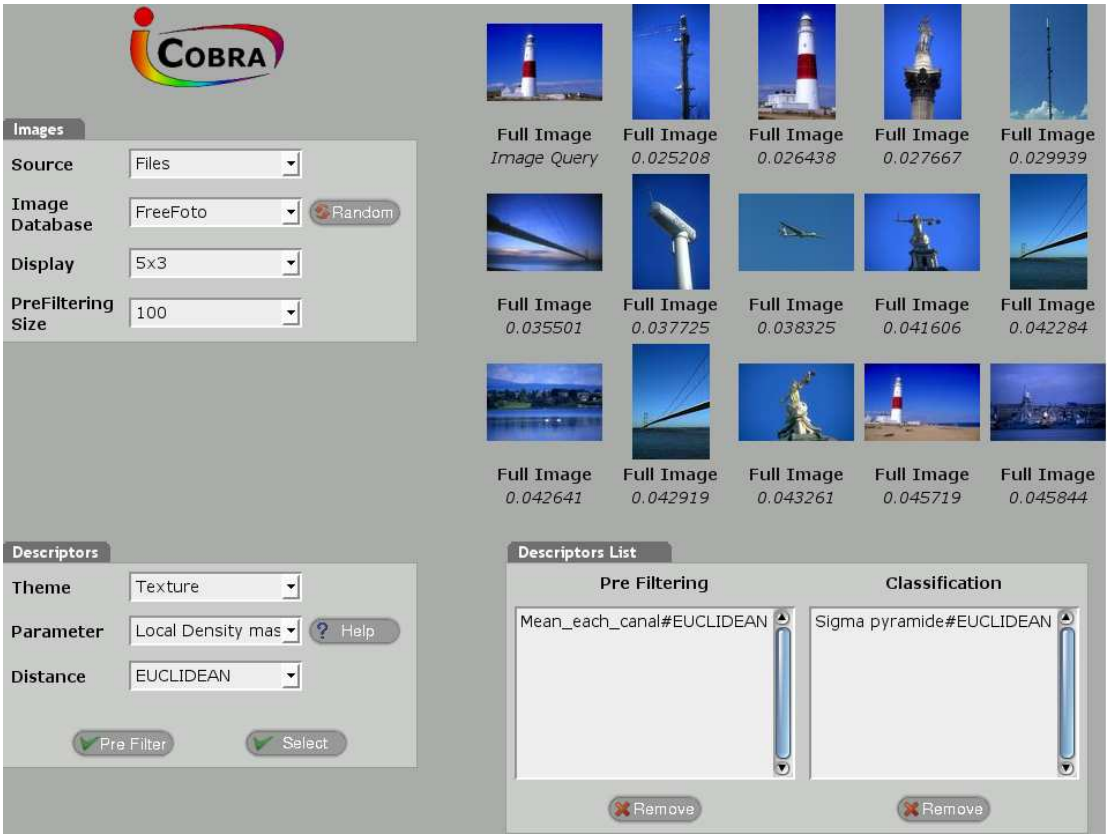


Fig. A.3 – *Utilisation d'*iCOBRA

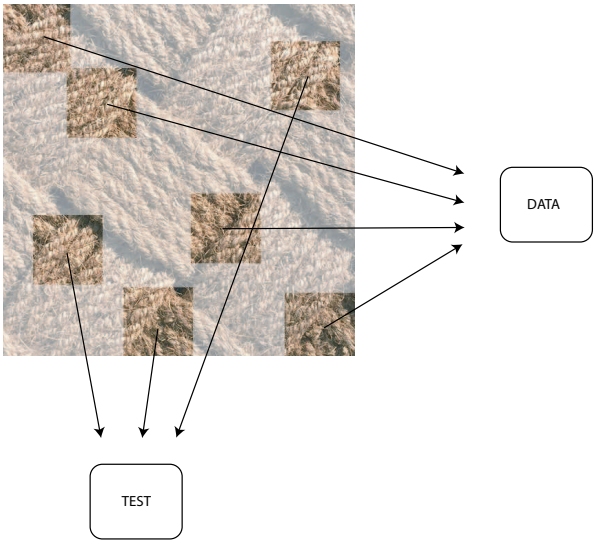


Fig. A.4 – *Création des ensembles DATA et TEST*

A.4 Exécution visuelle

L'objectif de cette partie est double. D'une part il s'agit de pouvoir tester des algorithmes de traitement d'images rapidement. Ainsi, une fois un ou plusieurs algorithmes sélectionnés, on peut naviguer et choisir d'exécuter ceux-ci sur des images bien précises. D'autre part, dans un but plus "pédagogique", il est possible pour chaque image de représenter son nuage colorimétrique en 3D via le logiciel ColorSpace. Ainsi l'utilisateur peut examiner simplement dans un grand nombre d'espaces le nuage couleur de n'importe quelle image de la base. La figure A.5 illustre cette interface.

A.5 Espaces couleur hybrides décorrélés

Cette interface permet de composer des espaces hybrides décorrélés à partir d'une ou plusieurs images. Ainsi il est possible de créer un espace couleur dans un contexte : la série d'images considérées. Avant toute chose, introduisons brièvement la notion d'espaces hybrides. Un espace hybride[Vandenbroucke *et al.*, 1998] est un n-uplet composé de combinaisons de composantes d'espaces pré-existants. Les espaces hybrides ont été introduits afin d'améliorer la capacité de l'espace à discriminer les couleurs et de réduire la corrélation entre les composantes. Nous proposerons après avoir introduit les espaces hybrides décorrélés une extension aux bases d'images. L'objectif de ces espaces décorrélés est de visualiser le nuage colorimétrique d'une image en

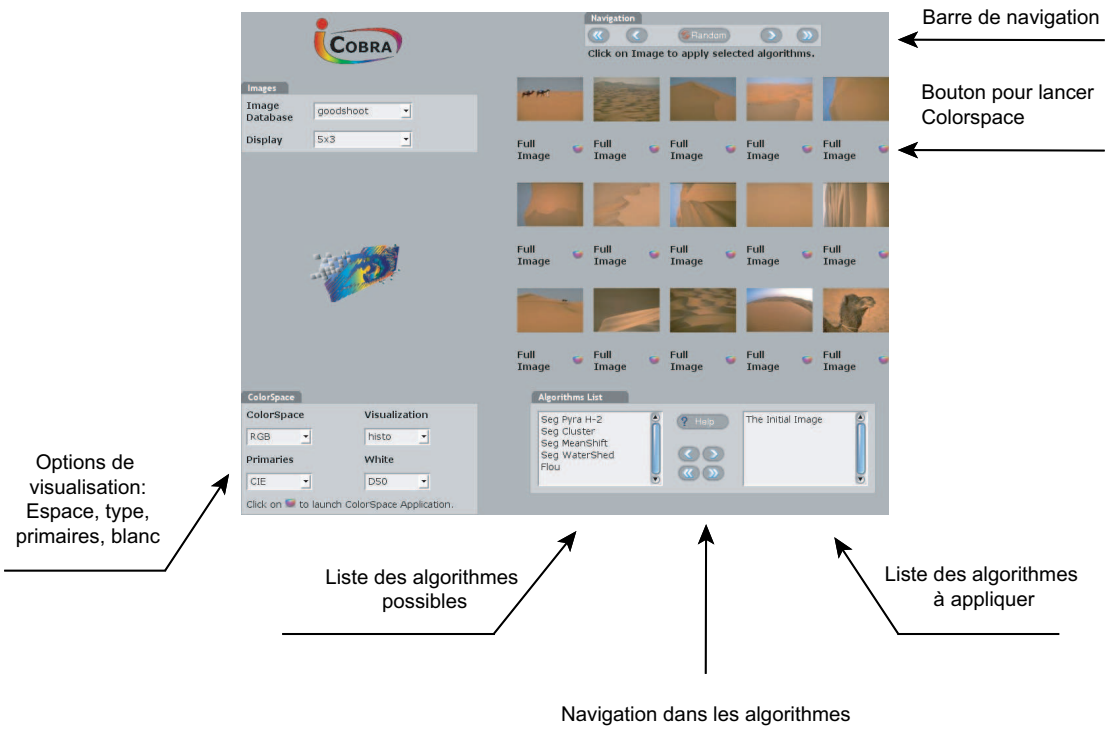


Fig. A.5 – *Interface d'execution visuelle*

fonction d'un contexte. Par exemple, la sélection d'une série de peintures d'un même auteur afin d'afficher dans un espace spécifique à cette série une peinture de cet auteur.

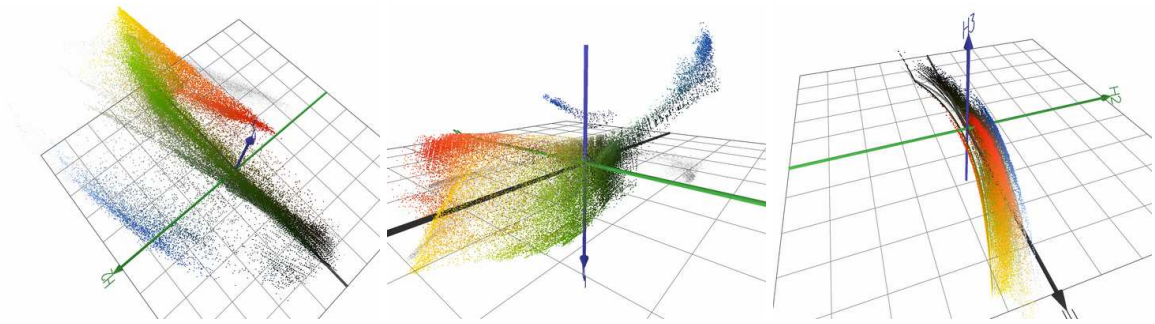
A.5.1 Espaces couleur hybrides décorrélés

La construction suivante [Colantoni et Trémeau, 2003] permet de construire un espace couleur hybride décorrélé.

1. Sélection manuelle de K composantes couleurs de différents espaces, chaque composante pouvant être pondérée.
2. Construction de l'image correspondante à K dimensions.
3. Construction de la matrice de co-variance (matrice $K \times K$).
4. Analyse en composantes principales[Duda *et al.*, 2001].
5. Sélection des 3 axes les plus significatifs. Ces 3 axes permettent de construire le nouvel espace couleur.

Les figures A.6 illustrent des représentations 3D de l'image Perroquet dans différents espaces hybrides A.6(b)[1], A.6(c)[1] et A.6(d)[2]. Par exemple, la figure A.6(d) montre la représentation du perroquet dans un espace généré uniquement à partir de composantes de type "luminances" de l'image. Le nuage correspondant est donc, comme attendu, très concentré. A contrario le nuage représenté en A.6(b) semble bien distinguer les trois couleurs principales de l'image.

(a) Perroquet



(b) (c) (d)

Fig. A.6 – *Exemples d'espaces hybrides*

A.5.2 Extension aux bases d'images

Détaillons maintenant les différents points amenant à un espace hybride à partir d'une base d'images. Pour cela, définissons avant toute chose les notations suivantes:

- S un ensemble d'images et S_l la l-ième image de taille $Taille\,(S_l)$.
- K l'ensemble des composantes couleurs sélectionnées et K_i la i$^{\text{e}}$ composante.
- $K_i^l(x,y)$ la valeur du pixel (x,y) de la composante K_i dans l'image S_l.

1. basé sur les compostantes R,G et B de RGB, X et Y de XYZ, L, M et S de LMS, cos de $SpectralPolar$
1. basé sur les composantes L^* et a^* de $L^*a^*b^*$, H et C de LHC, H et S de HSI
2. basé sur les composantes Y de xyY, L^* de $L^*a^*b^*$, L de LHC, I de HSI

Les matrices S et C sont alors définies par:

$$S_i^l = \sum_{xy \in S_l} K_i^l(x,y)$$

$$C_{ij}^l = \sum_{xy \in S_l} K_i^l(x,y) * K_j^l(x,y)$$

Pour une image S_l, la covariance correspond donc à :

$$Cov_{ij}^l = \frac{C_{ij}^l}{Taille\,(S_l)} - \frac{S_i^l}{Taille\,(S_l)} \times \frac{S_j^l}{Taille\,(S_l)}$$

Ainsi, en étendant la formule à N images, on obtient :

$$Cov_{ij} = \frac{\sum_{1 \leq l \leq n} C_{ij}^l}{\sum_{1 \leq l \leq n} Taille\,(S_l)} - \frac{\sum_{1 \leq l \leq n} S_i^l}{\sum_{1 \leq l \leq n} Taille\,(S_l)} \times \frac{\sum_{1 \leq l \leq n} S_j^l}{\sum_{1 \leq l \leq n} Taille\,(S_l)}$$

Comme précédemment, la construction de l'espace hybride est définie ensuite par les étapes :

4. Analyse en composantes principales.
5. Sélection des 3 axes les plus significatifs.

La figure A.7 montre comment sélectionner :

- Un ensemble d'images qui va servir de base à la génération de l'espace hybride.
- Un ensemble de composantes couleur support de l'espace hybride.
- L'image qui doit être représentée dans l'espace généré.

Au niveau du système proprement dit, comme décrit sur la figure A.8, les matrices C et S sont stockées dans une base de données afin de permettre une génération quasi immédiate de l'espace hybride.

La figure A.9 montre finalement quelques exemples d'espaces hybrides décorrélés appliqués aux bases d'images. Cet outil permet ainsi, en fonction d'une série d'images, donc un contexte bien particulier, d'afficher ou de traiter une image dans un espace issu de ce contexte. Par exemple, le nuage A.9f montre que, dans un espace généré par des images majoritairement bleues, le coucher de soleil est littéralement éclaté.

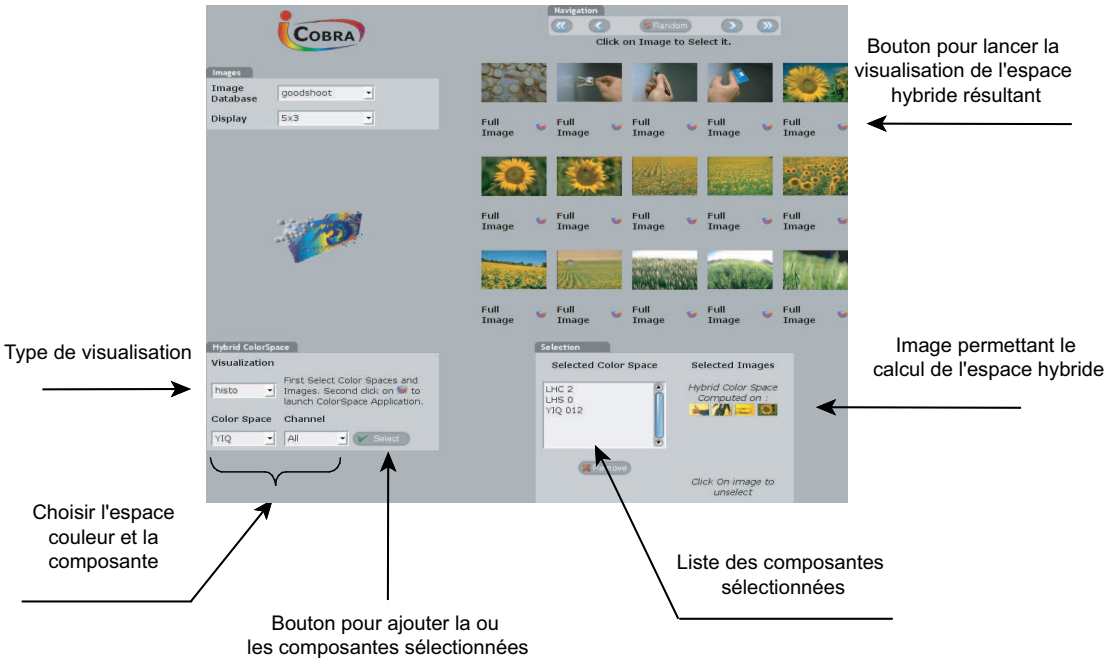


Fig. A.7 – *Espaces couleur hybrides décorrélés*

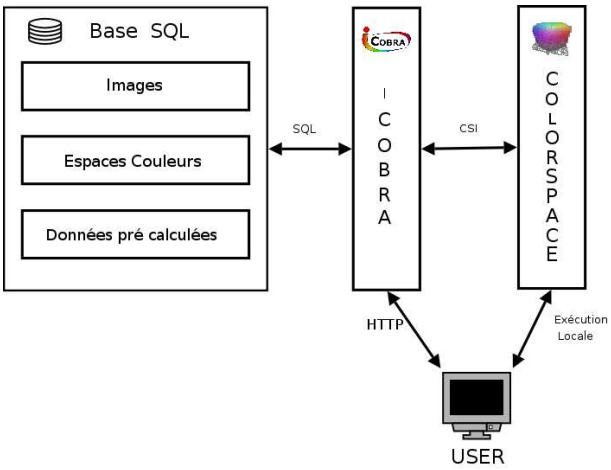
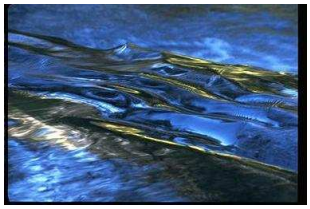
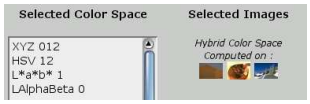


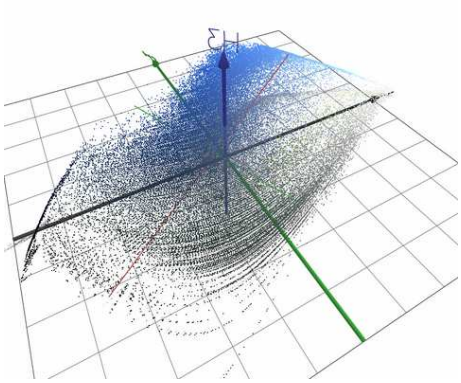
Fig. A.8 – *Utilisation dans* **i**COBRA



(a) Image originale

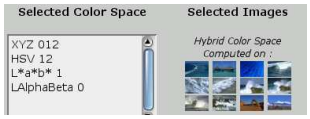


(b) Composantes sélectionnées

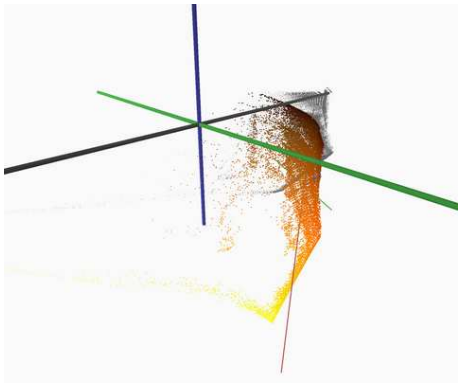


(c) Représentation dans l'espace hybride

(d) Image originale



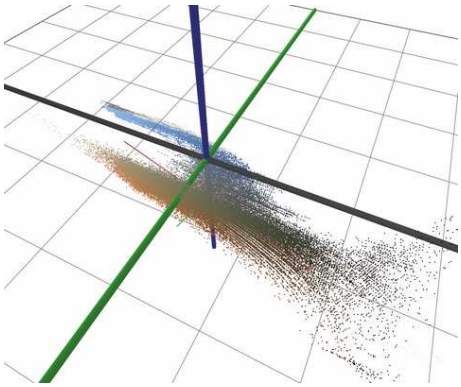
(e) Composantes sélectionnées



(f) Représentation dans l'espace hybride

(g) Image originale

(h) Composantes sélectionnées



(i) Représentation dans l'espace hybride

Fig. A.9 – *Exemples d'espaces hybrides décorrélés*

Chapitre B

LA CONSTRUCTION PYRAMIDALE

La pyramide classique est une suite de résolutions d'une image depuis l'image initiale jusqu'à des résolutions grossières, comme le montre la figure B.1. Son principal avantage est d'engendrer une hiérarchie entre les différents niveaux, chaque élément étant construit via des Fils au niveau inférieur et participant à des Pères au niveau supérieur. Explicitons maintenant précisément la construction pyramidale gaussienne utilisée tout au long de ce manuscrit.

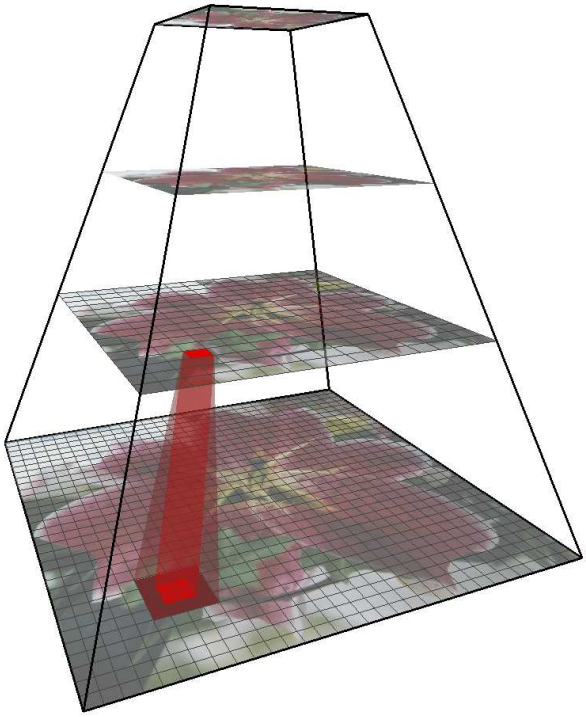


Fig. B.1 – *La Pyramide gaussienne avec recouvrement*

Soit G_k l'image de niveau k. G_0 est le niveau initial, ie l'image originale et G_k est une représentation à l'echelle 1/2 de l'image G_{k-1}. Le niveau de gris d'un pixel de la pyramide dite gaussienne est donc obtenu par la relation :

$$G_k(P) = \sum_{M \in Fils(P)} G_{k-1}(M) \cdot w(M)$$

où w est un filtre gaussien normalisé. De la même façon, on peut définir les images reconstruites :

$$B_k(P) = \sum_{M \in Pere(P)} G_{k+1}(M) \cdot W(M)$$

où W est le filtre inverse de w. On notera que G_k et B_k se retrouvent donc dans la même résolution.

Le choix des filtres et de l'ensemble des fils est modulable. Suivant la taille du masque, on construit donc des pyramides sans recouvrement (taille 2×2) ou avec recouvrement (taille $> 2 \times 2$) avec un filtre passe-bas normalisé. Durant tout ce manuscrit, les pyramides que nous considérons sont des pyramides avec recouvrement. Le masque utilisé, de taille 4×4, est un masque gaussien classique:

0.0169	0.0481	0.0481	0.0169
0.0481	0.1369	0.1369	0.0481
0.0481	0.1369	0.1369	0.0481
0.0169	0.0481	0.0481	0.0169

Tab. B.1 – *Le filtre gaussien* 4×4

Pour illustrer la construction, la figure B.2 montre les différents niveaux de la pyramide générée à partir de la première image.

Fig. B.2 – *Pyramide : un exemple*

La pyramide couleur s'obtient en appliquant la construction pyramidale sur les trois composantes R, G et B. Ainsi chaque niveau de la pyramide couleur sera la réunion des niveaux des pyramides de chaque composante couleur. Par essence même, cet outil présente l'intérêt majeur

d’évaluer conjointement l’information spatiale tout autant que l’information colorimétrique. Ensuite, comme le montre la figure B.2, les hauts niveaux de résolution moindre offrent une vision globale de l’image permettant une tâche d’extraction des régions principales plus aisée, alors que les bas niveaux comportent eux l’ensemble des détails nécessaires à une description précise de chacune d’elle.

Pour s’adapter à des images non carrées, ie des images de taille variable, la construction peut “perdre” une ligne et/ou une colonne à chaque nouveau niveau (de résolution divisée par 2 dans les deux directions par rapport au précédent). Tant que la construction n’est pas poursuivie jusqu’à des niveaux de résolution prohibitifs, où l’intégralité de l’information initiale est de toute évidence perdue, cette construction n’induit pas de pertes d’informations significatives.

ABSTRACT

The matter of this work is content based image retrieval and more precisely the contribution of the low level methods.

After having discussed the various existing approaches, we recall the semantic gap between the user expectations and what really the systems of research propose. Most of these approaches rely on a preliminary step of segmentation whose validity and robustness must be studied. Then we propose a protocol of evaluation and a practical example of benchmarks. The originality consists in not comparing a segmentation with a theoretical reference but judging its stability objectively.

The third part of this document introduces three specific contributions likely to improve the chain of research. Initially, a detector of blur allows to extract a meta-data carried by the image: the unblur regions, a priori of focusing. Secondly, we expose a descriptor based on the extraction of emergent areas using only the color criteria. This process, combined with adapted distances, may allow for example a color pre-filtering before the step of similarity research . Finally, we briefly introduce an algebra of histograms able as well as possible to exploit the information contained in this type of descriptors, via a specific query language.

KEYWORDS

Image retrieval, Segmentation, Benchmarking, Meta-Data extraction.

RÉSUMÉ

La thématique de ces travaux de thèse est la recherche d'images par le contenu et plus précisément l'apport des méthodes bas niveau.

Après avoir discuté des différentes approches existantes, nous rappelons le fossé sémantique entre les attentes de l'utilisateur et ce que proposent réellement les systèmes de recherche. La plupart de ceux-ci reposent sur une étape préalable de segmentation dont la validité et la robustesse se doivent d'être étudiées. Nous proposons alors un protocole d'évaluation objective et un exemple concret de mise en œuvre. L'originalité consiste à ne pas comparer une segmentation à une référence théorique mais à juger objectivement sa stabilité.

La troisième partie de ce document introduit trois contributions ponctuelles susceptibles d'améliorer la chaîne de recherche. Dans un premier temps, un détecteur de flou permet d'extraire une méta-information portée par l'image, les zones nettes a priori de focalisation. Ensuite nous exposons un descripteur basé sur l'extraction de régions émergentes sur le seul critère couleur. Cette extraction, conjuguée avec des distances adaptées, peut permettre par exemple un pré-filtrage couleur en amont de la phase de recherche de similarité proprement dite. Finalement, nous introduisons brièvement une algèbre d'histogrammes pour exploiter au mieux l'information contenue dans ce type de descripteurs, via un langage de requêtes spécifique.

MOTS CLÉS

Recherche d'images par le contenu, Segmentation, Évaluation objective, Extraction de métadonnées.